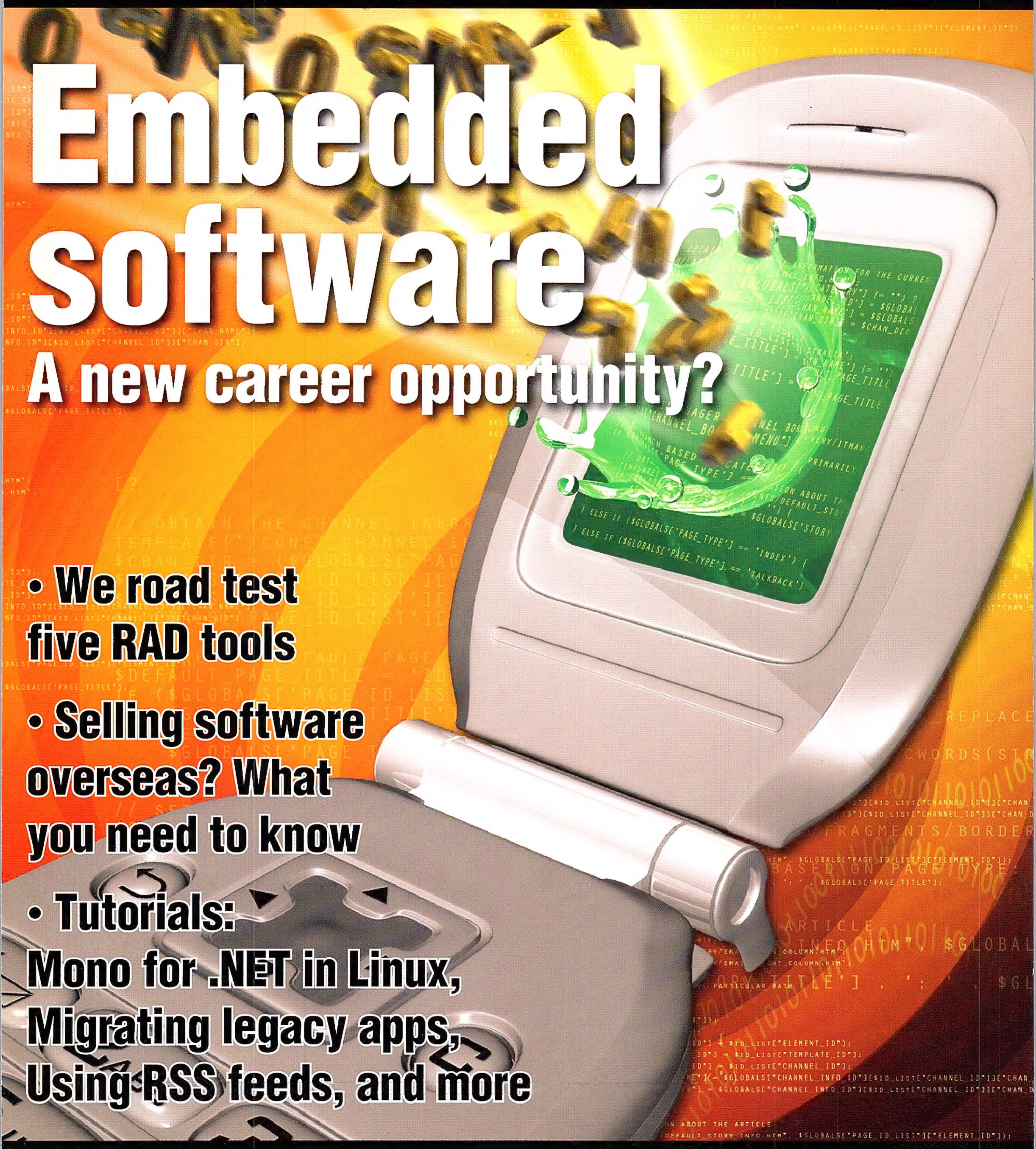
August 2004

BUILDER • MAGAZINE

By developers for developers

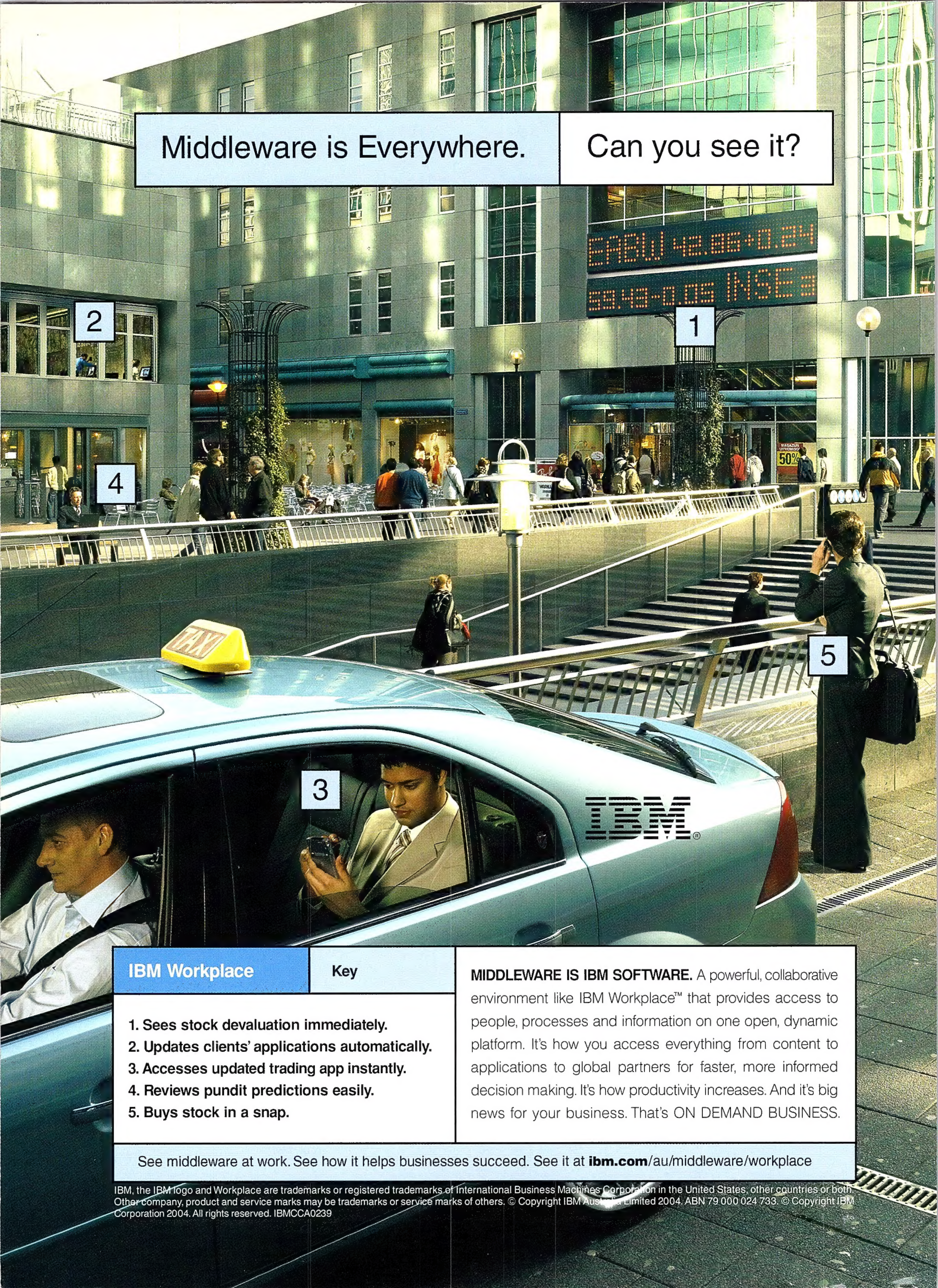


Embedded software

A new career opportunity?

- We road test five RAD tools
- Selling software overseas? What you need to know
- Tutorials: Mono for .NET in Linux, Migrating legacy apps, Using RSS feeds, and more

Brought to you by Technology & Business and www.builderau.com.au



IBM Workplace

Key

1. Sees stock devaluation immediately.
2. Updates clients' applications automatically.
3. Accesses updated trading app instantly.
4. Reviews pundit predictions easily.
5. Buys stock in a snap.

MIDDLEWARE IS IBM SOFTWARE. A powerful, collaborative environment like IBM Workplace™ that provides access to people, processes and information on one open, dynamic platform. It's how you access everything from content to applications to global partners for faster, more informed decision making. It's how productivity increases. And it's big news for your business. That's ON DEMAND BUSINESS.

See middleware at work. See how it helps businesses succeed. See it at **ibm.com**/au/middleware/workplace

Contents

BUILDER MAGAZINE

Commentary

Community Insight

Focus Feature

Program

Architect

Manage

Builder Magazine August 2004
Published by ZDNet Australia, a CNET Networks Company
ABN: 41 092 094 525 Phone (02)8514 9999 www.zdnet.com.au
Level 4, 45 Murray Street, Pyrmont, NSW 2009
Managing Director Wendy Hogan wendy.hogan@zdnet.com.au
Print General Manager Alison Paterson alison.paterson@zdnet.com.au

Advertising
Sales Manager Abigail Baker (02) 8514 9918 abigail.baker@zdnet.com.au
Senior Account Manager Tanya Charlton (02) 8514 9914 tanya.charlton@zdnet.com.au
Account Manager David McDowell (02) 8514 9962 david.mcdowell@zdnet.com.au
Account Manager Philip Duffield (02) 8514 9926 philip.duffield@zdnet.com.au
Production Manager Jenny Boshell jenny.boshell@zdnet.com.au

Builder AU Editorial
Editor Brendon Chase, brendon.chase@zdnet.com.au
Technical Editor David McAmis, david.mcamis@zdnet.com.au

Builder Magazine Editorial
Editorial Director Brian Haverty brian.haverty@zdnet.com.au
Editor Josh Mehlman josh.mehlman@zdnet.com.au
Assistant Editor Natalie Hambly natalie.hambly@zdnet.com.au
Senior Journalist Lisa Simmons lisa.simmons@zdnet.com.au
Art Director Craig Mackellar craig.mackellar@zdnet.com.au
Senior Designer Somme Sakounthong som@zdnet.com.au

Circulation
Circulation/Marketing Manager Rachelle Swiney rachelle.swiney@zdnet.com.au
Circulation Assistant Christian Ivers christian.ivers@zdnet.com.au

Builder Magazine is printed by Offset Alpine, Sydney.

Building on *Builder*

We've made a few changes to *Builder Magazine*, but a much more interesting change is coming up in a few months.

By Brendon Chase

Welcome to the August 2004 edition of *Builder Magazine*.

If you've already flicked through the magazine, your eyes haven't deceived you, we've made a few enhancements in this edition. Firstly we've decided to drop our mantra of one area of development per edition. We felt we were restricting the content too much and not giving other areas enough coverage. From here on in we aim have a wider breadth of topic areas per magazine that will hopefully appeal to everyone.

First up, we will be bringing you a new "road test" section managed by our Technical Editor, David McAmis. In this edition we will focus on rapid application development tools that claim to make your development efforts easier and faster.

The world of embedded devices is a mix and match of many development environments and methodologies, so this month Simon Sharwood takes a look at choosing embedded tools and platforms for developing embedded software.

The market for software in Australia is only a very small percentage of the overall market worldwide, a fact that drives many Australian developers to try and sell their wares in other countries. But to strike it rich in overseas markets there are a number of essential skills and tips you will need. Angus Kidman takes a look at some key considerations of taking your software offshore in part one of his two-part series.

Speaking of the next edition, be sure to look out for our special Builder Conference bumper issue. If you've missed the promos on Builder AU online about the conference, here's where the conference was up to at the time of writing.

The Builder team is currently working overtime to bring our first developers' conference in Sydney, which will run from October 5-7. We'll be flying in a few rock star developers and industry experts such as Richard Stallman of GNU fame, Ken Arnold who was one of the creators of Jini, Jonathan Wells from the mobile development team at Redmond, even a real rocket scientist!

Of course I cannot name all of our speakers in this editorial, but after receiving a huge response from our call for papers, I'm optimistic that the conference tracks are going to be top notch and will be appealing to developers from all walks of life.

The event will be broken up into five tracks that will cover over 30 unique breakout sessions for programmers, architects, and managers over the first two days. On the third day we will be offering two optional workshops, one introducing Extreme Programming and the other project management for developers, and you can be sure this editor will be sitting in the front row for that one!

Of course, like any conference it's not all talking shop; we'll be putting on a conference dinner that is going to feature comedian Adam Spencer as well as a host of fun activities.

Well that's enough of my blatant plug for now, be sure to check out *www.builderconference.com.au* for more up-to-date information on the conference agenda and social happenings and see you in October! ⬟

*Brendon Chase is editor of Builder AU (*www.builderau.com.au*). Contact him at* builder@zdnet.com.au.

> *"The market for software in Australia is only a very small percentage of the market worldwide, a fact that drives many Australian developers to try and sell their wares in other countries."*

2005 International

CES®

DEFINING TOMORROW'S TECHNOLOGY

PRODUCED BY CEA

THE HOTTEST TECHNOLOGY UNDER THE SUN.

Covering every major CE market, the International CES is the world's first look at the hottest technologies under the sun. Featuring 130,000 potential partners, competitors and market makers, and new products from 2,400 exhibitors. Make the connections that will make your year at the largest annual consumer electronics tradeshow on the planet. Watch your business grow. Register today at www.CESweb.org.

JANUARY 6-9, 2005 | LAS VEGAS

2005 International CES® | www.CESweb.org | grow

Developer spotlight: Jakob Nielsen

Recently renowned Web usability guru Dr Jakob Nielsen visited Australia to preach the word of good usability practices. We caught up with Nielsen to talk about the Web, open source software, the future of 3D interfaces, and mobile devices. By Brendon Chase

What is the most common question you've had on this tour about user interface (UI) design?
On the current tour, search was a big question people were asking about. One common question from every city was "How many people click on paid search ads?" The truth is, it is only a few percent but its still more than banner ads. More people are going to click on what are called organic search results.

Much of you fame has come around from your research in Web sites. As the browser becomes less of a focus, what is the future?
Well, I think there is still a lot of work to be done on this concept of Internet-based information. It's not so much based on the technology, it is the human needs. As in search, that is always going to be need. Then there is the next step of Internet-based applications, where more functions and features will appear. That is there now but it will become much more advanced in the future.

Is that because it's disguised as content?
It's true. Honestly, I don't think it means that means it's worth more to advertisers. It might be worth less to advertise this way because you don't necessarily want more clicks, you want high-investment-return clicks. You want people to click when they are the market for your product. You don't need to pay for regular browsers or free-loaders on the Web.

Many smaller Web sites' search functionality is inadequate, according to your latest research. Should developers be buying into proven search technology or building their own?
I don't care if they licence Microsoft, Google, Yahoo, or anyone as most of them are vastly better than the homemade search engines that most companies have. Each of these [search] companies has hundreds of top engineers working on improving their software so the probability that you can do it with one person is quite low.

What do you think of Web sites that run content management solutions where users get to choose their own design such as DotNetNuke?
It's a good idea. You really want to avoid having everybody design their own Web site because they usually do it wrong. While these templates might not always be the right ones, in principle you can put resources into making good templates and they can be reused and work well. Most intranets run this way and overall it works well.

There is much hype in the industry about free and open source software. From a usability standpoint how do you rate open source software compared to proprietary software?
Poorly, I'm sorry to say. I think the reason is that it's biased highly [towards] one specialised area which is the very technically based software such as sendmail, or Linux for IT systems administrators. But Linux for the average user or other open source solutions for someone who is not a geek rates particularly low.

The reason is, the motivation for open source is not because the person gets paid but the person gets prestige. The developers are designing for each other and they are so feature rich—geeks love features—and you get more prestige by adding features. For the average person fewer features is better and easier to understand.

The value systems are kind of opposite for what average users need and what open source developers want to do. As long as they are designing for other people like themselves it works quite well. But as soon as they try and design for the average person it breaks down and you have to have the more hierarchical project manager, you have to do user testing, and you have to do documentation. No developers like to write documentation.

The second problem is that open source when they turn to the general tools they tend to be in the line of "let's already implement what we already know" so they will take Microsoft Office and they will clone it. Since we've been criticising Microsoft for years for cloning Apple, it is only fair to criticise open source for cloning Microsoft. The point being that you don't move ahead but you have to do something new.

With a lot of commercial interests in these projects, do you think this will change?
I can certainly see that and I can also see [happening with] the office tools. We could look beyond [cloning] and there could be a

new set of productivity software. It could really be done another way. The real problem with Microsoft Office, besides being Microsoft, is that it is an office product based on paper. It comes out of research about how to do office automation.

In the new world what we need is not memos; we need to build Web sites and intranets so when you communicate in companies it is via a collaborative environments. Some areas where they have had success is in the Wikis and some of the blog software, but of course it is still quite primitive. But you could make something quite more interesting based on this on things like intranets.

What are your thoughts on 3D interfaces?

I think we should mainly work on the 2D design. There are cases where 3D works well like construction, engineering, medicine, and so on but most of the things we do with computers are more information-orientated. As long as we're stuck with a 2D computer screen that means the information needs to be in a 2D medium.

Do you think there will be more litigation in the area of usability, especially when it comes to discriminating against some users?

Probably yes, I'm not in favour of this however. I'm more in favour of the companies saying "we'll do it because we want to serve those customers". People tend to be loyal customers if they are discriminated against everywhere else they go, and you treat them well, then they tend to like you. But most companies don't seem to appreciate that so more likely there will more litigation.

In the world of mobile devices can you see history repeating itself with mobile applications like we did with Web sites when they were emerging?

Probably not. Every time we move into a new technology we see that it is used wrong and companies developing for it they do not recognise the special user limitations. We found that with WAP and it's sad to say I think its going to happen again. One of the downsides of the usability field is that people only tend to care about it after they have been burned.

In a recent article you talk about the next 30 years of computing. In that article you predict that software will protect against social engineering. How will software ever protect against that?

I know [laughs] that is why I put it into my 30-year prediction but I really think it is necessary. We have to find a way which we can help people against social engineering and it will take a long time.

Lastly if you were to give developers one piece of advice what would it be?

Look at users and study users. No matter what you are doing go and get hold of an actual real life end user and see what they do with the software. ⬟

BUILDER • CONFERENCE
5-7 October, Star City, Sydney

Builder Conference 2004

5-7 October, Star City, Sydney

The only independent Australian conference produced for developers by developers

The conference will be jam-packed with three days of keynotes, breakout sessions, and workshops presented by luminaries in the IT world, including keynotes from Richard Stallman, founder of the Free Software Foundation, and Ken Arnold, one of the inventors of Jini and key contributor to Java and a host of other developer tools and platforms.

Why Attend Builder Conference 2004?

At Builder AU, our philosophy is to facilitate the sharing of ideas and knowledge. Builder Conference provides an exceptional opportunity for developers, system architects and IT management to share their ideas and experiences both through presentations as well as informal networking opportunities.

- All major platforms and technologies covered. From Java to .NET, Windows to Unix to Linux, Open Source to Free Software, this is one conference that covers it all.
- Unbiased, real-world advice. There are no sponsored speaking slots at Builder Conference. All presenters are chosen strictly on merit.
- Network with your peers. You will have the opportunity to meet a wide range of your peers during the many social breaks during the event.

Want to know more?

See *www.builderconference.com.au* for all the details.

BROK-ERAGE
RE TIRE
CUSTOMER
JOIN PROMO
JOIN DATE
CLIENT NAME
REFERRAL
TRADE
ARBI TRAGE
TRADE USP
TRADE LIST
ADD RESS
LOGIN
USER ID
STATE
CITY
FUND
DATE
PX
QTY
EXEC INST
OPTION
OPEN CLOSE
DIFF
NAME
Microsoft
Your potential. Our passion.™

Embed me

Looking for a new career direction? Writing software designed to be embedded in an appliance, phone, or some other real-world device is a growth area, but has its own set of challenges.

By Simon Sharwood

It's getting harder and harder to avoid computers these days. The clock radio that wakes you up in the morning probably contains a computer, along with the microwave that warms your breakfast, the burglar alarm you set as you leave the house, the ticket reading machine at the train station, the lift that takes you into your office, and the security card reader at its door.

By the time you get home to pick up the messages on your digital answering machine, flop into your couch to watch a computer-powered DVD through your computer-powered TV, computers will have dominated your day without you ever noticing it.

For many developers this extraordinary prevalence of computers is also beginning to intrude into their conscious thoughts. Static wages, a glut of skilled coders, and the often-dull nature of much enterprise development has many wondering if their current skills would let them take on the new challenge of programming the embedded computers that surround us.

The good news is that the world of embedded development is already well within their reach and is becoming more so every day, opening new career options for those willing to learn some new tricks.

Most coders will be able to get a start in the embedded world if only because the languages used will be familiar. C and C++ are common and older languages like ADA and FORTRAN are still surprisingly prevalent, making existing development tools and skills more than viable for embedded development.

The process of software development should also be familiar. "At a really high level the steps involved in a project should be the same," says Malcolm Groves, Borland's regional product director for the Asia Pacific. "You gather business requirements, analyse how to translate them into software, run a testing phase, and then do it all again."

Yet these familiar coding and project management skills must adapt to some new and interesting challenges, foremost among which is the realtime capabilities of many embedded operating systems.

"Embedded operating systems need very high reliability," says Daya Nadamuni, a principal analyst with Gartner. "They must always be predictable: the system should guarantee you will get a response in a certain amount of time."

The qualities are important because embedded systems are often deployed in environments where milliseconds can literally be the difference between life and death. "A real time OS can never fail," Nadamuni says. "For things like avionics controls the result of failure is catastrophic."

Borland's Grove recalls a project for embedded systems to control a freight company's conveyor belts and package-sorting systems where the stakes were lower, but requirements just as sensitive.

"We had microseconds between when a barcode on a package was scanned and the embedded device issued the instruction about which chute it should go down," he says. "If you are writing that kind of software you need to know the operating system will send the message to the conveyor belt in time, because if it does not you have a problem."

Programmers for embedded systems

must come to grips with these issues, and the fact that addressing them requires work to understand how embedded OSes interface with the hardware in an embedded device.

"In embedded systems development there is quite a significant blurring of the line—you could argue there is no line—between hardware and software," says Daryl Wilding-McBride, practice lead for enterprise services at Object Consulting, who has coded extensively in the embedded and enterprise worlds. "With embedded systems you are much 'closer to the metal'. You are dealing with the pins on the chips and clock cycles and interrupts."

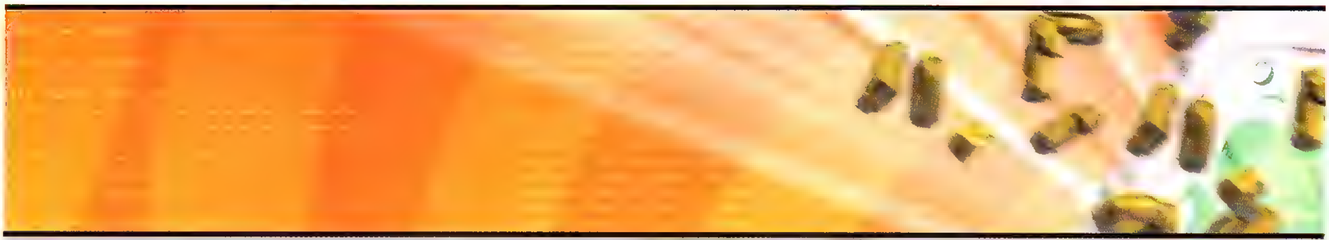
The technical parameters for these factors vary very highly between different embedded devices, which employ a vast range of processors and components that, according to Davyd Norris of IBM Rational, are a very significant barrier to mainstream developers hoping to cross over into the embedded world.

"The interesting thing about embedded development is the huge spread [of devices]," he says. "Embedded programmers tend to have highly developed but narrow skills specialised in specific architectures and microcontrollers." That need to specialise in different architectures—the venerable Z80 and Intel 6800 CPUs are still popular in embedded devices—has meant that many of today's embedded developers have a background in hardware design or engineering.

"An embedded developer used to be a low level programmer, often an older person brought up with assembler and C and their knowledge of the OS and the device is very tightly coupled," says Jason Tolley of Melbourne embedded development specialists ROK Technology.

While this trend often made for very good code, it also made for unwelcome complexity. Different architectures often demanded different development tools, sometimes with unique characteristics. Organisations developing for several different architectures could find themselves employing programmers whose skills were not suited to projects beyond their expertise, while also being asked to maintain expensive single-platform tools.

> *"Patching is hard when the embedded devices you programmed are installed down mines in Siberia and Western Australia."*



How to choose embedded tools

Choosing embedded software tools, such as assemblers, compilers, and linkers, can be quite daunting. Sometimes the decision is made for you; for instance, when there's only one tool-chain available, the semiconductor manufacturer sells it, and it costs a small fortune. You're forced to pay the big bucks and move on.

But how do you choose embedded tools when there are options?

The more you pay, the worse the tools

My first rule is to buy the cheapest tools that work. I don't mean that you should skimp on tools; it's just that the most expensive tools aren't necessarily the best. Actually, it's been my experience that the more you pay for tools, the worse they are.

This surprisingly makes sense: expensive tools are usually costly because a small user-base or a complacent vendor—or both—sustain them. That's a recipe for unreliability. (And guess who winds up finding the bugs?)

Count the users

I recently made the mistake of following this "cheap is best" advice too blindly. I was tempted to replace a $10,000+ software tool-chain (which wasn't fancy, except for the price) with some new alternative development tools that cost a few hundred dollars. The new tools promised to do the same job at a fraction of the price. At that price, we could afford to have a copy for each developer, instead of having a single tool tied to one PC.

I placed the order, the software arrived, and I duly put it through its paces. Several days later, I still couldn't persuade it to build a simple real-life project. The tool-chain was bug-ridden. The vendor fixed the bugs promptly but we found more—and so it continued. The low, up-front cost of a few hundred dollars was eclipsed by the cost of downtime we spent trying to get it to work. I gave up and returned to the mega-buck tools.

What happened? Though the tools were cheap, I overlooked (or chose to ignore) that the tools had very few existing users, just like some of the expensive counterparts. I was back to being a beta tester on someone else's project. I should have asked the vendor to supply references, ie, some real users who could speak about their experiences with the tools.

Don't be afraid to ask for references (and follow up on them) if you find yourself in a similar situation.

Companies that get it right

Some companies get things right. Microchip is a case in point; it offers good tools at reasonable prices.

However, most semiconductor companies don't get it. I dislike paying a premium for their tools just for the privilege of being able to design their hardware into end products. I also object to paying even more for dongles and sub-standard, glossy-looking text editors, when all I need are reliable command-line tools at a reasonable price.

Open source wins

The logical solution is to use freeware or open source tools when possible. These tools are free, you get the source code, and the tools usually have many users.

Picking the right embedded tools can be a difficult process, so take your time. The wrong choice can hit your project hard where it hurts: in bugs and missed deadlines.

—*David Brenan*

Those tools and people will remain for real-time projects, as their extreme demands cannot be compromised.

However, the big move in embedded systems is now towards more generic development tools capable of targeting more types of hardware to simplify development. Borland's approach has been to create tools that integrate the packages for developing with specific devices under one umbrella IDE. Microsoft's .NET Compact Framework and Sun's Java 2 Micro Edition (J2ME) represent the apex of that effort, as both allow developers to use everyday enterprise-class development environments to target diverse devices well beyond the desktop and promise to shift embedded development out of the hands of silver-haired veterans and into the mainstream.

"The .NET compact framework and the SDK architecture are similar between Windows XP and Windows XP Embedded," Tolley says. "You don't need to know the intricacies of the hardware. The learning curve is not that great," with the need to pause and reflect on the kind of code needed to work with the small memories and CPU capacities of embedded devices the prime consideration for those developers that do make the move.

The shift to embedded work will also require extra focus on certain aspects of software development. "The biggest challenge is debugging," Tolley says, as it needs to be extremely rigorous. "There are emulators but you still need to deploy [on an actual device] and the mindset around the process is different to the desktop environment," especially when the embedded device is intended for use in a car and a system crash at 100 km/h is likely to cause crashes of a much more serious nature.

A good debugging process is also important because many embedded devices are infrequently connected or operate in remote locations.

"Patching is hard when the embedded devices you programmed are installed down mines in Siberia and Western Australia," says Borland's Groves. "How often have you asked if the software on the set-top box is up to date? On the desktop Windows Update goes off and does it for you. It is very possible to cross over because the high-level steps are similar, but you have a whole bunch of new things to learn and to

> *"A real time OS can never fail. For things like avionics controls the result of failure is catastrophic."*

Meet the embedded software family

The embedded development world has some players most enterprise developers may never have encountered. Here's a guide to some of the most important players.

Wind River Systems. Wind River Systems is the world's largest embedded systems software company. The US-based company's VxWorks operating system is enormously popular and the company offers a vast range of tools to put it to work. NASA's Pathfinder probe used the company's wares, which are also available in versions tailored to various vertical industries.

www.windriver.com

MontaVista Software. MontaVista Linux is a version of the open source OS tailored to the needs of embedded developers. Available in "Professional," "Carrier-grade," and "Consumer Electronics" editions, the OS is accompanied by an IDE and other development tools.

www.mvista.com

QNX. QNX's Neutrino realtime operating system offers a true microkernel POSIX support and even symmetrical multiprocessing. The company aims its OS and accompanying Momentics development tools at automotive, medical, communications, defense, and control markets.

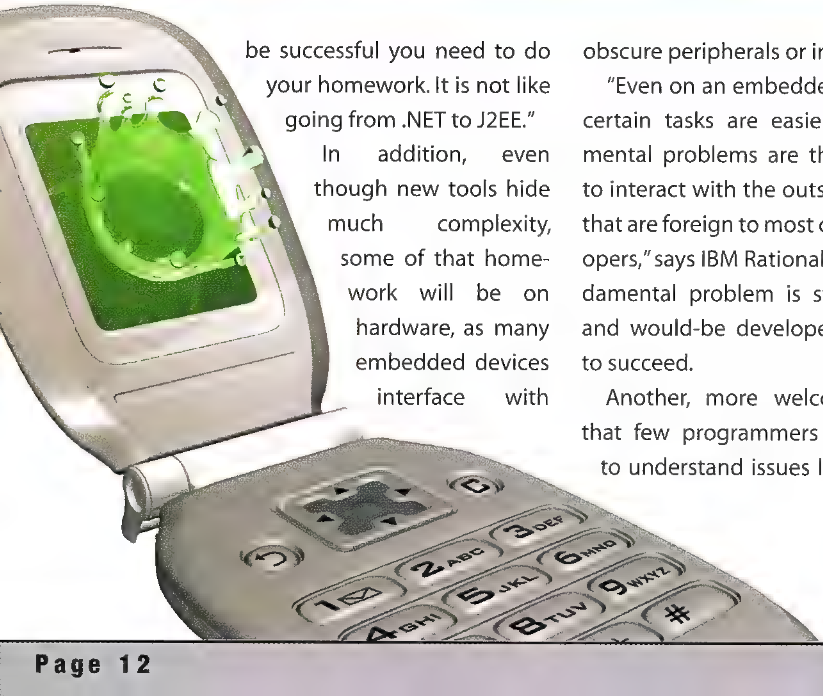
www.qnx.com

Green Hills Software. The well-regarded INTEGRITY realtime operating system is Green Hills Software's lead offering, with the royalty-free Velocity microkernel also a more than useful tool. Its MULTI development tool offers coders the choice of C, C++, or FORTRAN.

www.ghs.com

Mentor graphics. Mentor's products include software and hardware development tools, a combination it believes make it possible to develop embedded devices from the ground up faster than when each is developed independently.

www.mentor.com



be successful you need to do your homework. It is not like going from .NET to J2EE."

In addition, even though new tools hide much complexity, some of that homework will be on hardware, as many embedded devices interface with obscure peripherals or input devices.

"Even on an embedded PC architecture certain tasks are easier but the fundamental problems are that you still need to interact with the outside world in ways that are foreign to most commercial developers," says IBM Rational's Norris. "The fundamental problem is still input/output," and would-be developers must tackle it to succeed.

Another, more welcome, problem is that few programmers have are willing to understand issues like I/O, creating a shortage of programmers with the skills to embark on embedded projects. This shortage has already seen ROK's Tolley initiate a relationship with a local tertiary institution to encourage undergraduates to study embedded software development and adopt it as a career.

For developers looking for a technical challenge, a change of career direction or the chance of a fattened pay packet, embedding yourself in embedded computers therefore seems like a very sound idea indeed. ⬟

Testing, testing: the overlooked aspect of development

Software testers can be vitally important to the success of a project, so why don't they need to be accredited?

By Nathan Brumby

Until recently, there has been no distinction whether those testing software were suitably qualified and experienced to call themselves software test specialists or whether they were of a certain level of competence to participate in quality assurance and acceptance testing. With consumers becoming increasingly savvy, demanding best practise, the inaccuracies and shortcomings that came with ad-hoc guerilla testing, are now intolerable.

Despite being vitally important to the success of a product or project, the role of the software tester has typically been overlooked or, at best, undervalued by industry.

A lack of formally trained professionals and robust, standardised testing technology, performance metrics, methodologies, procedures and use of tools has contributed to high-cost problems such as increased software project failures (due to poor quality), increased software development expenditure, increased time to market due to inefficiencies, and increased market transaction costs.

The emphasis on software projects has traditionally been time to market, perhaps a driving force behind neglecting software testing. However, with their expectations rising, customers are demanding products that work first time around.

In November last year, Software Engineering Australia commissioned Sweeney Research to undertake a study into the attitudes and opinions of the Australian software industry. According to these results, 71 percent of Australian software-developing organisations believe there is room for improvement in the area of software testing. Overall, software testing ranked second in the list of issues most likely to keep people awake at night, after lack of business opportunities.

Nearly all respondents believed the planning process needed to be more formalised, but only 52 percent believed software testers need to be accredited.

Many software testers are insufficiently skilled or formally trained in the testing area. So what distinguishes the software test professional from the person who does testing?

Certification in the software test profession can help to ensure testing is conducted in a structured way. Where a person who does the job of testing can tell you something's broken, a software test professional can articulate his/her testing procedures to communicate the quality of the product to management, at any point in time, so that they can make informed decisions regarding the potential risk to the business. In turn, this ability to communicate testing procedure helps to make the testing activity more visible within the organisation.

As more coding and development work is sent offshore, the need for effective testing will increase. Testers are vital in ensuring effective application integration and conducting usability testing.

Software testing is a discipline that is only now beginning to gain the recognition it deserves in and outside of the software development space. Technically skilled testers are beginning to be seen as a valuable resource.

Certification tests the tester's skills and upon completion serves as evidence of a professional's ability to implement a formal standardised testing process and provides those who are serious about software testing with the recognition they deserve and helps to recognise testing as a discipline in its own right.

Nathan Brumby is CEO of Software Engineering Australia, www.seanational.com.au.

> *"Despite being vitally important to the success of a product or project, the role of the software tester has typically been overlooked or, at best, undervalued by industry."*

Totally RAD: we road test five IDEs

We get down and dirty with the most popular integration development environments to see how they stack up as rapid application development tools.

By David McAmis

It seems like every vendor with a software development tool or platform claims that it can be used for "Rapid Application Development" with little evidence to back it up.

What is RAD?

In addition to being a marketing buzzword, the term RAD is used to describe applications that can be designed and developed within 60-90 days. James Martin first explored the concepts of RAD while working at Dupont in the mid-eighties. It was there that he and Scott Shultz formalised a system of developing applications using a methodology he developed called rapid iterative production prototyping, that used flowcharting to design programs and applications.

Martin is considered to be the father of computer-aided design and his original work has been grown, augmented and expanded into the discipline we currently know as RAD. Over the years, RAD has grown to include a number of basic tenets for what defines a RAD project.

To start, a RAD project is characterised by the use of prototypes. A RAD prototype helps to jump start the design process and can demonstrate the look and feel of the application, as well as cutting down on the time required to gather requirements from users for the initial features and functionality.

The concept is that with a basic set of requirements from users, a developer can quickly build a prototype that the user can interact with and suggest features, enhancements, etc., usually in a workshop setting, as part of joint requirements planning (JRD) or joint application development (JAD) process. So any development tool or platform that is touted as a RAD tool should be able to facilitate taking user requirements from a JRD workshop and quickly creating an application prototype that can be reviewed and modified in a JAD workshop with users.

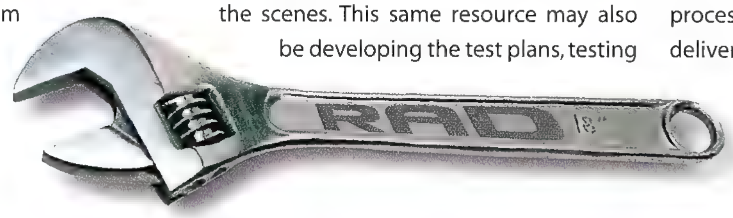
A good RAD tool should also provide developers with the tools to quickly add and remove features without requiring extensive re-writes, using a component-based architecture. In addition to changing user requirements during the course of a RAD project, most projects are "timeboxed", meaning that there is a set length of time that has been set for the completion of the project. Any features or functionality that is not able to be delivered within that set time frame are removed or deferred to future release.

A second attribute of applications created using a RAD methodology is that there are a number of developers who may be working on the application at the same time and these developers can fulfil a wide range of roles within their team. For example, you may have a developer who has also created the architecture for the application in question, as well as designing the user interface and the code behind the scenes. This same resource may also be developing the test plans, testing the application, writing documentation, and eventually training users. In a more formal project, these roles may be divided among multiple resources. In a RAD project, time and resource constraints often mean that these roles (and more) must often by undertaken by a single developer. For a IDE to support this facet of RAD, it must be adaptable to the different roles a developer must play and support each of these roles in turn.

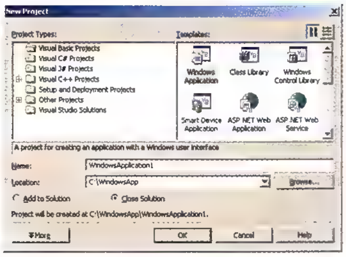
In addition to supporting the different roles within a team, a RAD tool must also be able to support the use of third-party components to deliver user requirements. In the debate over build-vs-buy, developers must be able to buy the components they don't have the time or inclination to build. For example if a developer is doing both the coding and user interface design for an application, they must be able to integrate components that could cut down the time required for either task (ie, code libraries, UI components, etc.)

Finally and most importantly, the litmus test for applications created using the RAD methodology is their fitness for a particular business purpose. In the normal phases of an application created using the proper software development lifecycle (SDLC), there would be a formal design process, where there are a number of deliverables during the cycle, including formal interviews, detailed design documents, semantic mapping between existing systems, process documentation, etc. that must be delivered.

In a RAD project, the primary



question at the end of the project is "Does this application suit the business process for which it was created?" If the answer is yes, then the project is considered a success. To that end, a RAD tool should provide the ability to quickly create an application that solves the business problem at hand. And while there are elements of the SDLC that may be included in a RAD project, this is not a primary concern. For example, with a true RAD tool, the ability to generate a process-flow diagram or database schema is not as important as delivering the functionality required by the business process.



A project for creating an application with a Windows user interface

Name: WindowsApplication1

Location: C:\WindowsApp

○ Add to Solution ● Close Solution

Project will be created at C:\WindowsApp\WindowsApplication1.

Figure 1: Microsoft Visual Studio.NET 2003

Testing the tools

When looking at the different development tools and platforms to road test, we walked through typical RAD projects including quick and simple data-entry forms, and tied to a simple database table on the back end. Following the RAD methodology, we built prototypes with each tool, noting where a tool provided efficiencies and where it could use improvement. We also looked at the features and functionality within the IDE that supported the tenets of RAD design and suitability for the required tasks.

Microsoft Visual Studio.NET 2003

Visual Studio.NET 2003 can be used to create Web and Windows applications for the .NET framework in a variety of languages, including C#, VB.NET, C++, J#, and other supported CLR-compatible languages, as shown in Figure 1. One of the distinct advantages of Visual Studio as a RAD tool is this multi-language support, enabling developers to build applications that have components written in different languages, enabling organisations to leverage the different skills that exist within the organisation.

For the quick creation of prototypes, Visual Studio also rates highly with a number of tools and components that can be used to quickly create data-driven applications. The tight integration with SQL Server is also a bonus, although developers who need to access information held on other platforms may find it cumbersome to connect to their underlying data.

That said, there are a number of database vendors who have tools that can snap in to Visual Studio, including Oracle and DB2 that make database integration less of an issue. In addition to database support, there is also a thriving industry around third-party components for Visual Studio, enabling developers to buy components for functionality they don't have the time or skill to build

The one major deficiency in considering Visual Studio as a RAD tool lies with the tools Microsoft has provided for teams of developers working on a single project. For versioning and source control, most Visual Studio developers will use Microsoft's Source Safe, which is difficult to use across multiple locations or work sites. In addition, Visual Studio does not have a lot of support for other developer roles (architect, tester, etc.) so you will need to employ other Microsoft or third-party tools to cater for the unique requirements of each of these roles.

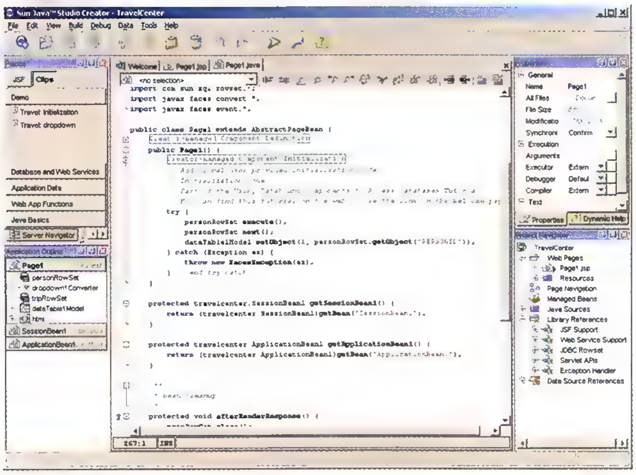
Microsoft has recently announced it will bundle a number of new tools in it's release of Visual Studio 2005 for architects, unit testing, and team development, but these tools are definitely missing from the current release and with a RAD project, you may not have the time to wait.

Sun Java Studio Creator

Sun recently released the first version of Java Studio Creator, which can be used to create JSP Web applications. The IDE itself, shown in Figure 2 is well organised but it does not have the sophistication of some of the other tools that have been on the market for longer.

Prototyping with Java Studio Creator is a cumbersome process, where even changing simple attributes like the font name and size can difficult. The one area where Sun has done its homework is in the development of multi-page applications, where you can drag and drop connections between pages, making it a breeze to quickly create links between them.

Luckily, what Java Studio Creator lacks for prototyping, it makes up for this in other areas, providing everything you need to develop and deploy Java applications, including a SQL database server, an application server, and the Java SDK. For developers who are looking to get started with Java development or projects where Java is the language of choice, this all-in-one approach can get a project



```
import com.sun.sql.rowset.*;
import javax.faces.convert.*;
import javax.faces.event.*;

public class Page1 extends AbstractPageBean {
    Creator-managed Component Definition
    public Page1() {
        Creator-managed Component Initialization
            [illegible]
        try {
            personRowSet.execute();
            personRowSet.next();
            dataTable1Model.setObject(1, personRowSet.getObject("PERSONID"));
        } catch (Exception ex) {
            throw new FacesException(ex);
        } // end try catch
    }

    protected travelcenter.SessionBean1 getSessionBean1() {
        return (travelcenter.SessionBean1)getBean("SessionBean1");
    }

    protected travelcenter.ApplicationBean1 getApplicationBean1() {
        return (travelcenter.ApplicationBean1)getBean("ApplicationBean1");
    }

    /**
     * Bean cleanup.
     */
    protected void afterRenderResponse() {
```

Figure 2: Sun Java Studio Creator.

underway quickly.

However, for a RAD project that requires a team of developers to work together on a single project, Java Studio Creator is not a good fit—you are better off going with Sun's Java Studio Enterprise product, which is an integrated suite of tools more suited to large development teams. But for a small development team that needs to get underway quickly, Java Studio Creator may fit the bill.

As for third-party components, there are a number of Java components out there, but few (if any) that integrate with the Java Studio Creator IDE. As the tool matures you may see more support for an integrated toolset, but in this first release, it doesn't quite stack up against other more mature tools.

BEA Web Logic Workshop 8.1

WebLogic Workshop 8.1 is BEA's entry into the RAD tool stakes and can be used to create Web applications, Web services, JSPs, portals, EJBs, and process workflows. When you open WebLogic Workshop for the first time, you will notice a similar look and feel to the other RAD tools reviewed, but Web Logic Workshop provides a number of tools that make it easier to create prototypes of Web applications.

To start, the data integration is tight and allows you to quickly create data-driven Web pages and sites, with a number of pre-configured controls that will allow you to easily bind to a data source and display, edit, and update the data. This includes controls that you can use to control repeating elements and paging, which in other tools you may end up coding by hand. You can also use page flows within WebLogic Workshop to separate presentation, business logic implementation, and navigational control, as shown in Figure 3.

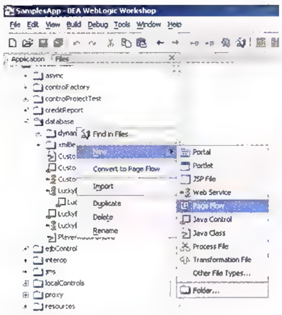
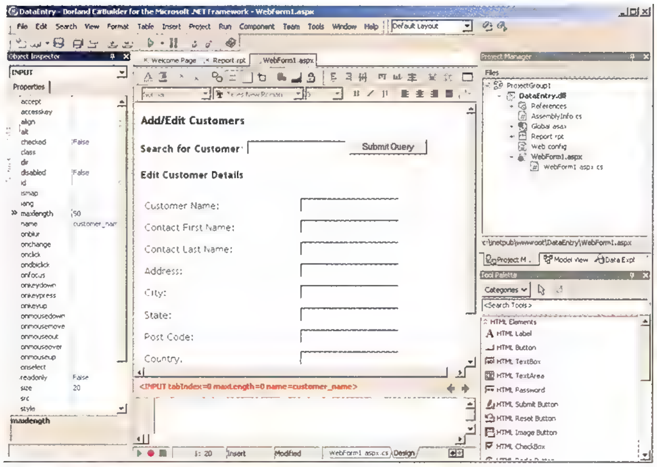


Figure 3: BEA WebLogic Workshop 8.1.

There is also a thriving community of third-party providers for components that can extend the BEA WebLogic Platform. And while there are not as many components available compared to those available for Visual Studio, most of the major categories of third-party tools (user interface, charting and graphing, additional functions, reporting, etc.) are well represented.

The user interface within the WebLogic Workshop IDE is also intuitive and easy to learn, even if you are new to BEA tools or Java development. In addition, building and testing your applications from within the IDE is a quick and straightforward process. This means that you can quickly iterate through multiple versions of an application without having waiting for the build process.



```
<INPUT tabIndex=0 maxLength=0 name=customer_name>
```

Figure 4: Borland C# Builder.

For developers who are new to JSP development, BEA WebLogic Workshop provides the easiest way to quickly develop JSP applications without having to know a lot about the underlying framework. For more experienced Java developers, the IDE also provides a number of advanced features that will also make them feel at home and in control of the underlying components.

For interoperability, you can use BEA WebLogic Workshop to create Web services that can be consumed by other platforms or languages (including .NET) without having to do any manual intervention or modifications to the Web services themselves.

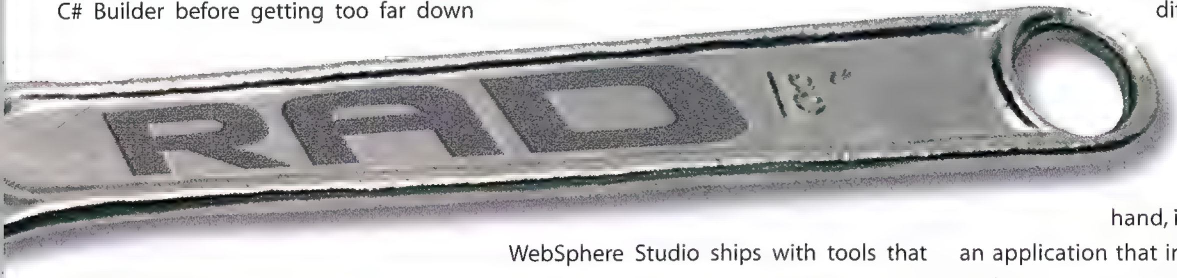
Borland C# Builder

Borland's C# Builder is the wildcard for .NET development, providing an alternative IDE for creating applications for the .NET framework. Borland has had some experience in creating IDEs for various languages and platforms and that experience shines through within C# Builder, shown here in Figure 4.

To start, prototyping with C# Builder is painless, with a clean, consistent IDE that makes it easy to quickly create and format page elements. The user interface itself is similar to the UI found in Visual

Studio and developers familiar with that environment will have no trouble with productivity in C# Builder. For compatibility between tools, projects from C# Builder can be exported to Visual Studio-compatible projects.

In addition sharing a similar appearance to the Visual Studio IDE, C# Builder can also leverage most of the third-party components and tools that are available for Visual Studio, even if they were not written for C# Builder (although check with your third-party vendor to ensure that these components are supported for use with C# Builder before getting too far down the development path.)



For building component-driven applications, the ace that C# Builder has over other RAD tools is its interoperability with J2EE applications through Janeva, Borland's object request broker, which is an implementation of Visibroker for the .NET framework. Using Janeva you can leverage the existing J2EE applications within your organisation, which could potentially save time during a RAD project.

Team-oriented development is also well supported within C# Builder with support for source control and versioning. There are also a number of tools and technologies that have filtered down from other Borland products that make C# Builder a versatile tool for modelling and building applications, supporting a variety of developer roles (architect, modeller, developer)

IBM WebSphere Studio Application Developer 5.1.2

IBM's foray into RAD tools comes in the form of IBM WebSphere Studio, which is based on technology from the Eclipse project. WebSphere Studio, shown in Figure 5, is best suited for an experienced development team, with a number of advanced features that are sure to confuse novice programmers and those developers new to Java development.

The trade-off is that WebSphere Studio can be considered a "power tool" for RAD development. When creating prototypes of Web applications you can quickly layout the entire site map for your application using the Web Site Navigation tools and then fill in the rest later. In addition, WebSphere Studio ships with tools that support JavaServer Faces components, so if you have some experience with Faces you can speed consist development.

And for development spread across a large team, WebSphere Studio also uses Rational ClearCase LT for version control and the integrated UML tools cater for architects who may also be working on the project. IBM has done a good job of integrating the technology from Rational into its development tools, which is a bonus for developers looking for a single-vendor (or even single-product) solution.

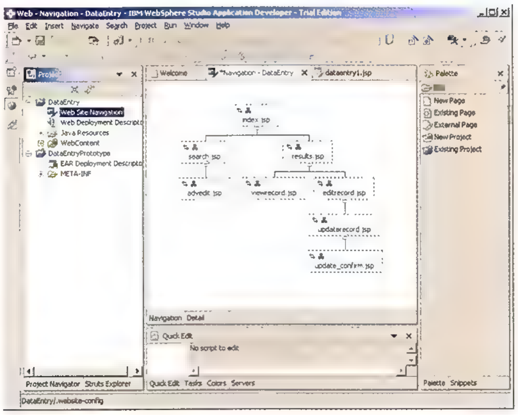


Figure 5: IBM WebSphere Studio Application Developer.

Summary

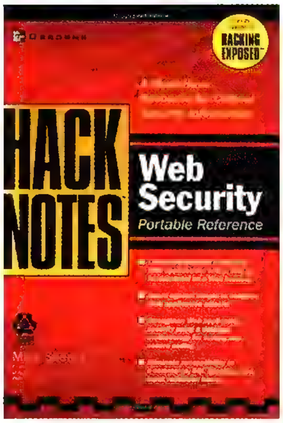
If you had to pick one tool from this group that catered for all of the tenets of RAD, you would be hard pressed. Each of the tools we looked at have their own unique strengths and areas where they are better suited to a particular situation. For example, if you are considering a RAD project where you need to quickly create prototypes using a team of developers who have experience with different CLR languages, Visual Studio may provide the best tool for that environment.

On the other hand, if you need to build an application that integrates both .NET and J2EE components and are willing to standarise on a single language, C# Builder may be the best tool for the job. And the plethora of third-party components and add-ins means that you can buy commonly used components rather than build them yourself.

From the Java side of things, it is two-horse race between the tools from BEA and IBM. The distinct advantage that IBM has comes from the Rational tools that IBM has integrated into its product—in a large team environment this can speed development and time between prototype iterations.

So in the end, it depends on the developer resources you have on hand and their skill sets, your preferred platform and required features. While none of the IDEs we have looked at present a complete RAD solution, they do provide enough support for RAD to make them all contenders for your next RAD project. ⬟

Book reviews



Hack Notes: Web Security Portable Reference

By Mike Schema

Publisher: McGraw-Hill/Osborne

The *Hack Notes Web Security Portable Reference* is a handy book for Web developers not familiar with Web security. In a relatively short 170-odd pages it fits quite a lot of introductory and reference material.

The first part of the book deals with hacking techniques and defences. This goes through various techniques hackers use, points of exploitation, and how to distinguish different types of attacks such as session hijacking and SQL injections.

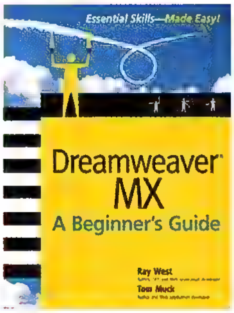
The second part of the book looks closely at platform level and programs that support a Web application. It highlights the use of some handy vulnerability tools to help developers uncover any nasties and to strengthen or harden their platform.

The last part of the book deals with Web server analysis and tips for secure coding. The tips are brief, but if you have limited expertise it is a satisfactory first read.

The quick reference guide inside the book makes it quite a useful tool.

Hack Notes, Web Security Portable Reference will appeal to newbie Web developers wanting to learn a quick overview of Web security and the experienced Web developer who wants a quick reference book nearby "just in case".

—Brendon Chase



Dreamweaver MX 2004—A Beginner's Guide

By Tom Muck and Ray West

Publisher: McGraw-Hill/Osborne

In the early days of the Internet, visual HTML editors Dreamweaver and FrontPage in the wrong hands were responsible for some pretty horrendous Web sites.

Dreamweaver MX 2004 has come a long way since its early releases and now provides enormous power to Web developers wanting to create simple static Web pages and up to highly complex Web applications.

This beginners' guide is a great start for budding Web developers. This book teaches new developers about more than just the product.

The book explains the fundamentals of the Internet, TCP/IP, installing a testing environment, user interface design, the differences between server-side technologies, security basics, an SQL primer, and debugging your code; all essential skills a good Web developer will need to grasp.

The book not only explains the theory, it digs a little deeper and demonstrates to readers how they can get their hands dirty in some Web applications on a variety of platforms and technologies using Dreamweaver MX 2004. Unfortunately, the book doesn't come with a CD bundled.

If you are new to Web development this book is an essential read before picking up a tool like Dreamweaver, in fact, if you are going to use Dreamweaver, I beg you read this book!

—Brendon Chase

Beginning Red Hat Linux 9

By Sandip Bhattacharya, Pancrazio De Mauro, Mark Mamone, Kapil Sharma, Deepak Thomas, Simon Whiting, Shishir Gundavaram

Publisher: WROX Publishing

There aren't many developers or IT professionals out there who have not had some experience with Windows, either as a desktop or server platform. *Beginning Red Hat Linux 9* has been written to leverage that knowledge and provide a common frame of reference for learning about Linux.

The book itself is well organised and written, although readers with some experience with Linux may find the emphasis on working with the desktop a bit frustrating (as opposed to jumping right to a command line). The book assumes no prior knowledge of Linux and does an excellent job of explaining the basic design of the operating system.

The chapters and topics are laid out well and follow a natural progression. Towards the end there is a chapter on "Managing Your System with Perl" that seems out of place in a beginner title; there are a number of good books on Perl available that would be more appropriate.

As a Windows developer who is learning about developing applications on Linux, this book was essential reading for getting started and is an invaluable reference.

—David McAmis

Drilling down into ADO.NET version 2.0

The new data access model in the upcoming release of the .NET framework could mean rethinking the way your applications handle data connections.

By Daniel L Fox

Version two of the .NET Framework is on the horizon. Now is the time to start thinking about ADO.NET v2 and its impact on how you'll develop applications.

With the first, public beta version of Visual Studio .NET 2005 (formerly christened "Whidbey") now in the hands of developers, it's time to start thinking about your applications and how they might be affected as you move to this new version.

Although the move from version 1.0 to 1.1 of the .NET Framework was trivial and involved mostly bug fixes, performance enhancements, and the integration of previously separate technologies such as the ODBC and Oracle .NET Data Providers, version 2.0 changes the story for data access. It includes a host of new features, some of which may cause you to rethink how data is accessed and manipulated in your applications.

Providing a wider view of data

Before delving into the specific features of ADO.NET v2.0, let us preface the discussion by noting that one of the overall design goals of this version was to allow a higher degree of interoperability between data accessed relationally, accessed as XML, and accessed as custom objects. Since these three make up the ruling triumvirate in representing data in an application, ADO.NET v2.0 was designed to make it easier for developers to use the appropriate model when desired within and across applications.

For example, in applications built using a service-oriented architecture (SOA), persistent business data will often be manipulated relationally; data that represents the process and encapsulates business rules will be manipulated as objects; and message and lookup data that must be serialised for transport will be handled as XML. To present the new features, We've factored them into two broad buckets: the new features that provide this wider view of data and the features that enhance or extend the relational paradigm.

Widening the .NET

There are two primary new features you'll want to explore in the area of extending your ability to handle data. Let's take a look at each.

ObjectSpaces

This technology was previewed several years ago at Microsoft's Professional Developer Congerence and will now be released in Visual Studio 2005. Simply put, ObjectSpaces provides an object-relational mapping layer in the System.Data.ObjectSpaces namespace, which instantiates and populates custom objects from a relational database. This works through XML metadata stored in a mappings file that is passed to the constructor of the ObjectSpace class, which maps relational objects to .NET objects and relational types to .NET types.

The programming model supports queries (ObjectQuery) and maintains sets of objects in memory (ObjectSet), access to streams of objects (ObjectReader), and even lazy loading of objects to improve

Listing A: The ObjectSpaces programming model

```
// Create the mappings
ObjectSpace oa = new ObjectSpace(mappings-file,
connection);

// Query the data
ObjectQuery oq = new ObjectQuery(Product,
"category='Equipment'");
ObjectReader or = oa.GetObjectReader(oq);

// Traverse the data
while (or.Read())
{
Product p = (Product)or.Current;
Console.WriteLine(p.Name);
}
```

Listing B: The XmlAdapter and XML Views

```
// Set up the connection and query
SqlConnection con = new SqlConnection(connection-string);
XQueryProcessor xq = new XQueryProcessor();
xq.XmlViewSchemaDictionary.Add(“name”, new
XmlTextReader(“mappings-file”));
xq.Compile(…);

// Set up the datasource
XmlDataSourceResolver xd = new XmlDataSourceResolver();
xd.Add(“MyDB”, con);

// Configure the XmlAdapter
XmlAdapter xa = new XmlAdapter(xd);
XPathDocument xp = new XPathDocument();

// Execute the query and populate the document
xa.Fill(xp, xq.XmlCommand);

// Navigate the document…
XPathNavigator xn = xp.CreateXPathNavigator();

// Or edit the document and change the data
XPathEditor xe = xp.CreateXPathEditor();

// Set the schema and update the database
MappingSchema ms = new MappingSchema(“mappings-file”);
xa.Update(xp, ms);
```

performance (ObjectList and ObjectHolder). Listing A shows an example of how the programming model looks.

Although in the current release, ObjectSpaces works only with SQL Server 2000 and SQL Server 2005 (the upcoming release of SQL Server, formerly codenamed "Yukon", more or less synchronised with the release of Visual Studio 2005), this technology will be extended to access other relational stores in the future. ObjectSpaces is ideal when you want to represent your data using a domain model and encapsulate business logic as methods on your custom objects, since it will save you from writing the tiresome code needed to load from and persist your objects to a relational data store.

SQLXML and XmlAdapter

Although the ADO.NET DataSet has always included the ability to load data as XML and serialise its contents as XML, the translation between the two ways of representing data always included some tension. For example, in order for the XML to load into a DataSet, its schema couldn't be overly complex, and it needed to map well into the relational DataTables of the DataSet.

Although DataSet support of XML has been enhanced in version 2 to allow loading of XML with multiple in-line schemas, loading schemas with repeated element names in different namespaces, and loading/serialising directly from DataTable objects, the data must still be relational in nature to work with the DataSet. To overcome this, version 2 includes the System.Xml.XmlAdapter class. This class is analogous to the DataAdapter classes in that it is a liaison between a data source and a representation of the data, but is used to query and load XML from an XML View into an XPathDocument object (called XPathDocument2 in the alpha; however, that will be renamed to XPathDocument before release).

XML Views allow relational tables (in SQL Server only) to be mapped to an XML schema via mappings files; they are the core component of the SQLXML 3.0 technology, once provided separately from the .NET Framework but now migrated into ADO.NET v2 (including the ability to bulk-load XML into SQL Server) in the System.Data.SqlXml namespace. Using this approach, you can provide a set of XML Views for your SQL Server data, query the data with the XQuery language using the new XQueryProcessor class and the Fill method of the XmlAdapter, and manipulate the data using the XPathDocument, XPathEditor, XPathNavigator, and XPathChangeNavigator classes.

The changes are written to SQL Server by calling the Update method of the XmlAdapter, which relies on the XML View to write the SQL statements to execute using a mappings file. The advantage of this approach is that you can treat your SQL Server data no differently than other XML data stores and can take advantage of the full fidelity of the XML when making changes. Listing B provides a simple example of the programming model.

Of course, XML Views simply provide the mapping of the data to and from SQL Server. If you're not using SQL Server, you can still take advantage of the substantial changes to XPathDocument (that will supersede and make obsolete the XmlDocument class) and its related classes to more easily query, navigate, and edit XML that

Listing C: An asynchronous command

```
// Set up the connection and command
SqlConnection con = new SqlConnection(connection-string);
SqlCommand cm = new SqlCommand(SQL statement, con);
cm.Open();
cm.BeginExecuteNonQuery(new AsyncCallback(DoneExecuting),
null);

// Thread is free, do other things
...

// Callback method
private void DoneExecuting(SqlAsyncResult ar)
{
   int numRows = ar.EndExecuteNonQuery(ar);
   // print the number of rows affected
}
```

you load from other sources.

For example, you can use a new XmlFactory class to create a related set of XmlReader, XmlWriter, and XPathNavigator classes for an XML document. These classes now support the ability to read and write .NET types to and from XML documents. And, of course, performance has improved for reading and writing with XmlTextReader and XmlTextWriter, and when using XSLT.

Extending the relational paradigm

The second broad set of changes relates to those made in ADO.NET v2.0 to enhance relational database access. We've organised these into changes that all developers can take advantage of, regardless of the underlying database you write to and regardless of those that will require SQL Server 2000 or 2005.

- **Provider factory.** Although the design of .NET Data Providers is based on a common set of interfaces and base classes, in v1.0 or v1.1, Microsoft did not ship factory classes to help developers write polymorphic data access code. As a result, developers did so on their own.

In version 2, ADO.NET includes factory classes inherited from System.Data.Common.DbProviderFactory to create the standard connection, command, data reader, table, parameter, permissions, and data adapter classes; these help you write code that targets multiple databases. A factory is accessed using the GetFactory method of the DbProviderFactories class and can be configured in the application's configuration file using the DbProviderConfigurationHandler.

- **Asynchronous data access.** Commands executed by ADO.NET in version 1.0 using the ExecuteNonQuery, ExecuteReader, and ExecuteXmlReader methods of SqlCommand were synchronous and would block the current thread until the results were returned by the server. In v2.0, each of these methods includes both Begin and End versions to support asynchronous execution from the client's perspective.

This technique employs the familiar asynchronous programming model using the AsyncCallback delegate in .NET, and so includes the SqlAsyncResult class to implement the IAsyncResult interface. While this feature works only for SqlClient at the moment, look for it to perhaps be extended to other providers before the release. Listing C shows an example of setting up an asynchronous command. (Note that the SqlAsyncResult class is not included in the alpha at this time, so the code will not execute.)

- **Batch updates.** In version 1.0, a DataAdapter always sent changes to rows one at a time to the server. In version 2.0, the DataAdapter exposes an UpdateBatchSize property that, if supported by the data provider, allows changed rows to be sent to the server in groups. This cuts down on the number of round-trips to the server and therefore increases performance.
- **Data paging.** In both SqlClient and OracleClient, the command object now exposes an ExecutePageReader method that allows you to pass in the starting row and the number of rows to return from the server. This allows for more efficient data access by retrieving only the rows you need to display. However, this feature reads the rows currently in the table, so subsequent calls may contain rows from the previous page because of inserts, or from the latter pages because of deletes. It therefore works best with relatively static data.
- **Binary DataSet remoting.** Version 2.0 now allows DataSets to be serialised using a binary format when employing .NET remoting. This both increases the performance of remoting data between .NET applications and reduces the number of bytes transferred.
- **DataSet and DataReader transfer.** In version 1.1, you could only load a DataSet from a DataAdapter. But in version 2.0, you can also load one directly using a DataReader and the Load method. Conversely, you can now generate a DataTableReader (inherited from DbDataReader) with the GetDataReader method in order to traverse the contents of a DataSet. This feature makes it easy to load a DataSet and view its data.

Climbing Yukon

In this category are the new features of ADO.NET v2.0 that relate directly to the release of SQL Server 2005:

- **MARS.** Multiple active result sets (MARS) allows you to work with more than one concurrent result set on a single connection to your database server. This can be efficient if you need to open a SqlDataReader and, during the traversal, execute a command against a particular row. MARS allows both commands to share the same SqlConnection object so that a second connection to SQL Server is not required.
- **Change notification.** One of the most interesting new features of SQL Server 2005 is its ability to support notifications. ADO.NET v2.0 includes programmatic support for this feature by including a SqlNotificationRequest object that can be bound to a SqlCommand.

When data returned from the command changes in the database, a message is sent to the specified notification queue. ADO.NET code can then query the queue either by using an asynchronous query that blocks until a message is sent or by periodically checking the queue using new Transact-SQL syntax.

To make this feature even easier to work with, a SqlDependency class that sets up an asynchronous delegate is included. This will be called when the data changes, and it can be used like other dependencies in conjunction with the ASP.NET caching engine. Listing D shows an example of using a SqlDependency object.

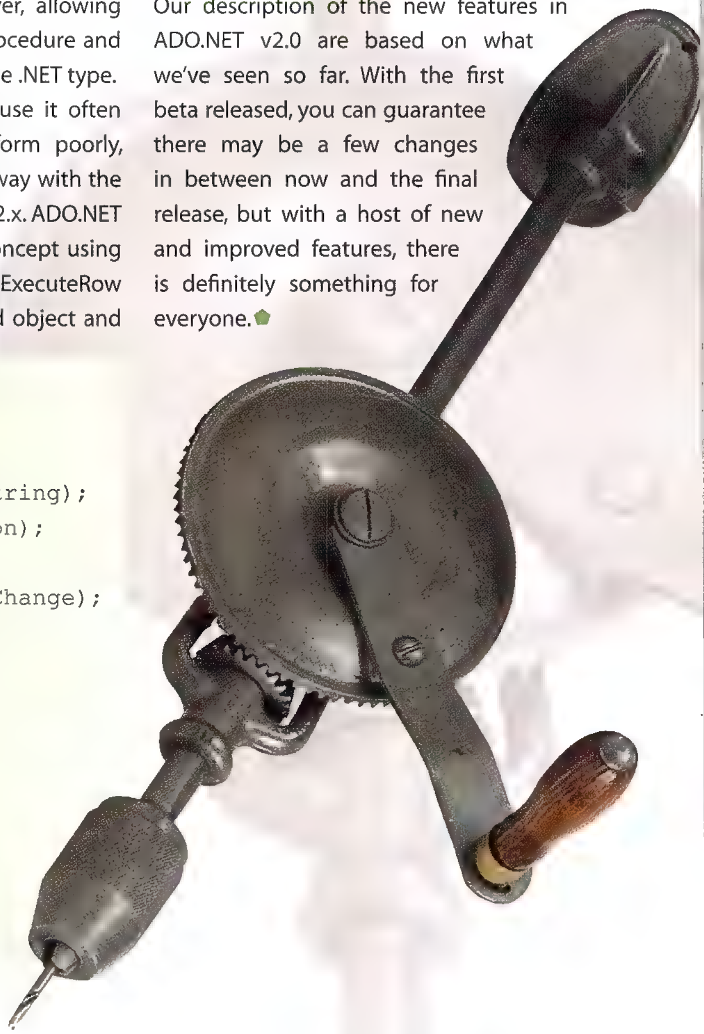
- **New data types.** ADO.NET v2.0 supports the full set of SQL Server 2005 data types, including XML and User Defined Types (UDTs). This means that columns in SQL Server 2005 defined as XML can be retrieved as XmlReader objects, and that UDTs can be passed to stored procedures and returned from queries as standard .NET types. This allows your applications to work with data as fully formed objects while interacting with the database using the objects. This feature can be used profitably when writing managed code that runs in-process in SQL Server, allowing both the managed stored procedure and the client code to use the same .NET type.
- **Server-side cursors.** Because it often caused applications to perform poorly, ADO.NET v1.0 and v1.1 did away with the server-side cursors for ADO v2.x. ADO.NET v2.0 now reintroduces the concept using the ExecuteResultset and ExecuteRow methods of the SqlCommand object and the SqlResultset class.

The SqlResultset class offers a fully scrollable and updateable cursor that can be useful for applications that need to traverse a large amount of data and update only a few rows. Although this feature can be used from client applications such as ASP.NET, it is mainly intended for use when writing managed code that runs in-process within SQL Server 2005 in the form of stored procedures.

- **Bulk copy.** Although not restricted to SQL Server 2005, ADO.NET v2.0 now allows programmatic access to the BCP or bulk copy API exposed by SQL Server. This is done using the SqlBulkCopyOperation and SqlBulkCopyColumnAssociator classes in the System.Data.SqlClient namespace.

More to come

Our description of the new features in ADO.NET v2.0 are based on what we've seen so far. With the first beta released, you can guarantee there may be a few changes in between now and the final release, but with a host of new and improved features, there is definitely something for everyone.



Listing D: A notification using SqlDependency

```
// Set up the connection and command
SqlConnection con = new SqlConnection(connection-string);
SqlCommand cm = new SqlCommand(SELECT statement, con);
SqlDependency dep = new SqlDependency(cm);
dep.OnChanged += new OnChangedEventHandler (HandleChange);
SqlDataReader dr = cm.ExecuteReader();

// Process the data
...
private void HandleChange (object sender,
SqlNotificationEventArgs e)
{
  // A change has been made to the data
  // Inspect the type of change using e.Type
}
```

Parsing RSS feeds with PHP

RSS is becoming a popular format for sharing content such as online news and blogs with the rest of the world. Here's how you can incorporate those feeds with your Web site.

By Shylon Ray Hunter and David McAmis

RSS 2.0 is an XML vocabulary that provides a means for describing news and events so they can be shared across the Web in a simple and standardised way. Sites such as *Builder AU* and *ZDNet Australia* offer feeds of their news articles that you can incorporate into your own Web site.

Creating the XML parser

So how do you incorporate RSS into a PHP Web site? Since RSS is XML-based, we can use PHP's XML libraries to handle the parsing of RSS data elements. The PHP XML extension is implemented via Expat, an XML parser written in the C language. Let's get started by looking at the code below:

```
<?php
// Create an xml parser
$xmlParser = xml_parser_
create();
// Free xml parser form
memory
xml_parser_free( $xmlParser );
?>
```

In this example, we created a basic XML parser and then freed up the XML parser resource. This is a very simple example from which we'll build a small RSS application. The first line calls the xml_parser_create function, which does pretty much what its name implies—creates an XML parser. The last line of code, the call to the xml_parser_free function, frees up the XML parser from memory. Now that we have the basic creation and removal of the XML parser, we can proceed to greater things.

Element handlers

In this next example shown in Listing A, we've created some functions and made various XML function calls to set handlers in the parser. The three functions are used to handle specific events that occur when an element starts or stops and when there is character data.

The first function, startElement, will handle all the opening XML tags. Basically, when the parser comes to a new element (XML tag), the startElement function will receive that tag name as one of its parameters. The second parameter is the parser resource, and the third parameter contains the attributes associated with the corresponding element (tag). The attributes are contained in an array that we've named $attrs. As the parser passes the tags to this function, they're printed out again as XML tags (not a very productive function, but it illustrates what we want to do).

The endElement function does the same thing as startElement, except it deals with the closing tag of the XML elements. Not to be too exciting, we only print out the tag name again as an XML closing tag. Before this function is called by the parser, charElement is called after startElement because the character data is inside of the start and end elements. In the charElement function, again we're only printing out the data to the user. These functions play a vital role in the processing of XML elements, and you should familiarise yourself with how the parser works with each one. Though we're only using functions in this simple example, you have the ability to create a class, with its associated functions, to deal with all your XML parsing needs. This gives flexibility and modularity to your applications.

Now that the element functions are defined, there are still some missing pieces. The parser needs to know which functions will be handling which elements. To specify which ones will do what, the xml_set_element_handler function will set the correlation between the parser and the handling function. The function takes three arguments: parser, start element handler, and end element handler. The last two arguments are strings that contain the names of the functions the parser will use as user callback functions. Instead of strings, an array containing an object reference and a method name can be used to assign the associated callback function to theparser.

When we set the xml_set_character_data_handler function, we're causing the XML parser to send all character data to the specified user function. This function accepts two arguments: The first is the parser resource, and the second is the user-defined callback function that will receive the character data from the XML parser.

The user callback function must also accept two arguments: parser resource and a data string. The parser resource is the parser defined by the xml_parser_create function. The xml_set_character_data_handler can also accept an object reference similar to that described above instead of the user callback function.

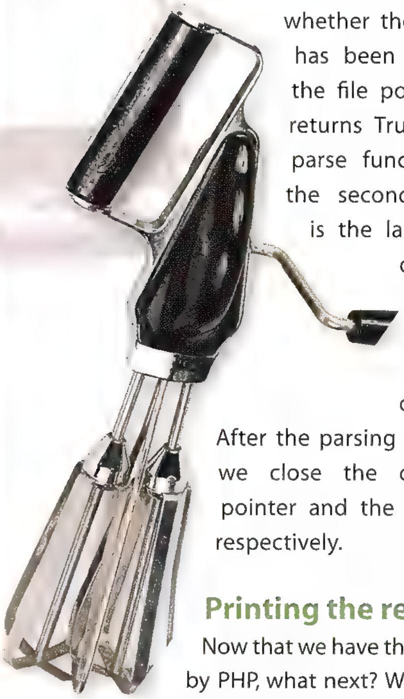
Listing A

```
<?php
$GLOBALS[‘title’] = false;
$GLOBALS[‘link’] = false;
$GLOBALS[‘description’] = false;
$GLOBALS[‘item’] = false;
$GLOBALS[‘titletext’] = null;
$GLOBALS[‘linktext’] = null;
$GLOBALS[‘desctext’] = null;

function startElement( $parser, $tagName, $attrs ) {
// By setting global variable tag name I can determine which tag I am currently parsing.
switch( $tagName ) {
case ‘ITEM’:
$GLOBALS[‘item’] = true;
break;
case ‘TITLE’:
$GLOBALS[‘title’] = true;
break;
case ‘LINK’:
$GLOBALS[‘link’] = true;
break;
case ‘DESCRIPTION’:
$GLOBALS[‘description’] = true;
break;
}
}

function endElement( $parser, $tagName ) {
// By noticing the closing tag, I can print out the data
// that I want.
switch( $tagName ) {
case ‘TITLE’:
if( $GLOBALS[‘item’] == true ) {
echo “<p><b>” . $GLOBALS[‘titletext’] . “</b><br/>”;
}
$GLOBALS[‘title’] = false;
$GLOBALS[‘titletext’] = “”;
break;
case ‘LINK’:
if( $GLOBALS[‘item’] == true ) {
echo “<a href=\””. $GLOBALS[‘linktext’] . “\”>View Article</a><br/>”;
}
$GLOBALS[‘link’] = false;
$GLOBALS[‘linktext’] = “”;
break;
case ‘DESCRIPTION’:
```

(continued next page)

Now that we have everything set up and defined, we can move on to actually opening and parsing the RSS file. We use the fopen function to open and set a file pointer to the RSS resource, in this case the Builder AU features feed. If the script can't open the file pointer, it simply stops. After we open the file pointer, we use a while loop to read in the RSS file. As we loop through the file data, we parse the XML data with the xml_parse function.

The xml_parse function takes three arguments; the first two are required while the last argument is optional. The first argument needs to be the parser resource; the second needs to be the data that is to be parsed. In the example in the listing below, we set the third argument to the feof function to test whether the end-of-file has been reached by the file pointer. If this returns True, the xml_parse function knows the second argument is the last chunk of data to be parsed for that given document or RSS file. After the parsing is complete, we close the opened file pointer and the XML parser, respectively.



Printing the results

Now that we have the RSS parsed by PHP, what next? We can set up the functions to print out only the information that we want out of the RSS file. The code in Listing A is a simple example of how to set up the functions to print only the title, a link to view the article link, and description. In the last example, you can get a feel for how we've set up the functions to work together to display specific elements that are desired for output to the user.

Listing A *(continued)*

```
if( $GLOBALS[‘item’] == true ) {
echo “ ”. $GLOBALS[‘desctext’] . “</p>”;
}
$GLOBALS[‘description’] = false;
$GLOBALS[‘desctext’] = “”;
break;
}
}

function charElement( $parser, $text ) {
// Verify the tag that text belongs to. I set the global
// tag name to true when I am in that tag.
if( $GLOBALS[‘title’] == true ) {
$GLOBALS[‘titletext’] .= htmlspecialchars( trim($text) );
} else if( $GLOBALS[‘link’] == true ) {
$GLOBALS[‘linktext’] .= trim( $text );
} else if( $GLOBALS[‘description’] == true ) {
$GLOBALS[‘desctext’] .= htmlspecialchars( trim( $text )
);
}
}

// Create an xml parser
$xmlParser = xml_parser_create();
txml_set_element_handler( $xmlParser, “startElement”,
“endElement” );

// Set up character handler
xml_set_character_data_handler( $xmlParser, “charElement”
);

// Open connection to RSS XML file for parsing.
$fp = fopen( “http://www.builderau.com.au/feeds/
features.htm”, “r” )
or die( “Cannot read RSS data file.” );

// Parse XML data from RSS file.
while( $data = fread( $fp, 4096 ) ) {
xml_parse( $xmlParser, $data, feof( $fp ) );
}

// Close file open handler
fclose( $fp );

// Free xml parser from memory
xml_parser_free( $xmlParser );
?>
```

We'd like to think that not all perfect matches are made in heaven.

Crystal Reports 10

Which edition of Crystal Reports® is right for you?

	Report Author/IT Editions		Bundled Developer Editions		Full Developer Editions	
	Standard	Professional	.NET Edition[2]	Java™ Edition[3]	Developer	Advanced
Report Creation						
Visual report designer for rapid data access and formatting	●	●	●[1]	●[1]	●	●
Customisable templates for faster, more consistent formatting	●	●			●	●
Repository for reuse of common report objects across multiple reports[4]		●			●	●
Data Access						
PC-based and Microsoft® ODBC/OLE DB for MS Access and SQL Server	●	●	●	●	●	●
Enterprise database servers (ODBC, native)		●	●[1]	●[1]	●	●
Custom, user-defined data through JavaBeans™				●	●	●
Custom, user-defined data through ADO and .NET			●		●	●
Report Integration						
Report viewing APIs (.NET and COM SDKs)			●		●	●
Report viewing APIs (Java SDK)				●	●	●
Extensive report viewer options (DHTML, ActiveX, Java Plug-in, and more)					●	●
APIs for run-time report creation and modification						●
Report Parts for embedding report objects in wireless and portal apps	●	●			●	●
Report Deployment						
Crystal Reports components for report viewing, printing, and exporting:						
a) Java reporting component				●	●	●
b) .NET reporting component			●		●	●
c) COM reporting component					●	●
Full featured report exporting		●			●	●
Report server (Crystal Enterprise Embedded deployment license)						●

[1] Limited functionality. [2] Bundled with Microsoft® Visual Studio® .NET and Boland® C#Builder™
[3] Bundled with BEA WebLogic Workshop™ and Boland® JBuilder®. [4] This feature is available on the Crystal Enterprise CD, included in the Crystal Reports 10 package.

Perfect matches can be made here too. In order to quickly determine which Crystal Reports® best suits your project requirements, we've provided this basic feature chart. Crystal Reports® 10 simplifies the process of accessing, formatting, and tightly integrating data into Windows and web applications via an enhanced designer, flexible data connectivity options, and rich Java™, .NET, and COM SDKs.

To learn more about Crystal Reports 10, compare over 150 different features across versions, or to access technical resources like the Developer Zone and evaluation downloads, www.businessobjects.com/v10/anz.

For your nearest reseller please call **1800 634 088**

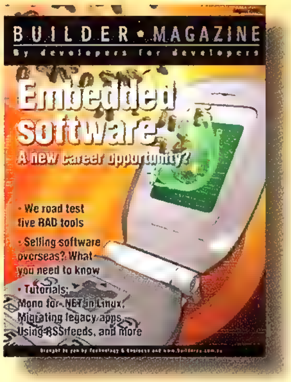
BUSINESS OBJECTS®

The business intelligence standard.

BUILDER • MAGAZINE

By developers for developers

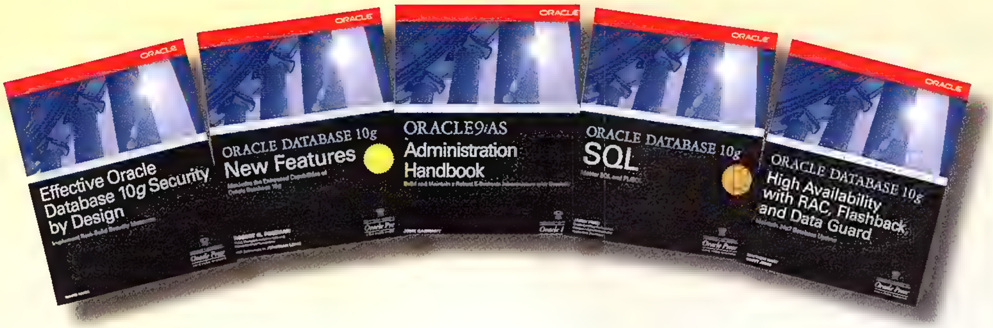
Apply to receive Builder Magazine free and have the chance to win one of five McGraw–Hill IT book packages worth over $2,000!


BUILDER • MAGAZINE
By developers for developers
Embedded software
A new career opportunity?
• We road test five RAD tools
• Selling software overseas? What you need to know
• Tutorials: Mono for .NET in Linux, Migrating legacy apps, Using RSS feeds, and more

Builder Magazine is the only magazine written by developers for developers. Utilising the information and feedback gathered from Builder AU, Builder Magazine's content influences and shapes the enterprise IT infrastructure.

Builder Magazine goes beyond the code to educate and inform the enterprise software community not just about programming, but architecture and management issues.

Builder Magazine delivers this information in an unbiased, vendor agnostic environment in a printed format, always available at the readers' fingertips for easy referral.


Effective Oracle Database 10g Security by Design
ORACLE DATABASE 10g New Features
ORACLE9iAS Administration Handbook
ORACLE DATABASE 10g SQL
ORACLE DATABASE 10g High Availability with RAC, Flashback and Data Guard

(From left to right)

Effective Oracle Database 10g Security by Design; By: David Knox; R.R.P: $99.95

Oracle Database 10g New Features; By: Robert Freeman; R.R.P: $59.95

Oracle Application Server 10g Administration Handbook; By: John Garmany Donald Burleson; R.R.P: $79.95

Oracle Database 10g SQL; By: Jason Price; R.R.P: $79.95

Oracle Database 10g High Availability with RAC, Flashback & Data Guard; By: Matthew Hart and Scott Jesse; R.R.P: $99.95

BUILDER ◦ MAGAZINE

By developers for developers

Mc Graw Hill Osborne

www.osborne.com

To apply for a free Builder Magazine subscription simply write to subs@zdnet.com.au and tell us in 20 words or less why you should receive a package of McGraw-Hill IT books.

Using Mono for .NET on Linux

Learn how to get Mono up and running for deploying .NET applications to Linux and other platforms.

By David McAmis and Rohan Gilchrist

The Mono Project has to be one of the most eagerly awaited open source projects for developers, bringing the .NET framework to Linux and other platforms, opening up a range of possibilities for application deployment.

The final version of Mono version 1.0 was recently released (see sidebar next page), but for we were only able to review the Beta3 version in time for publication. Mono Beta 3 (like all betas) should only be used for testing purposes only and is not recommended for production use. In this article we are going to look at how to get Mono up and running on Linux, and how to deploy your first .NET application to the Mono platform.

If you are Windows developer who is considering using Linux to host your .NET applications, the Mono beta provides a good way to test and tweak your applications before making the move.

For this walk-through we are going to use Red Hat 9.0 as our platform to install and configure Mono.

If you don't have a spare computer or server to load RedHat, you can turn your existing machine into a dual-boot machine but it may be easier to use a virtual machine like VMWare but make sure you read through any installation notes for installing Red Hat.

Once you have Red Hat up and running, you will need to download Mono from the the Go-Mono Web site. (*www.go-mono.com*) There are different versions depending on the operating system you are using, including:

- Red Hat 9.0/x86
- Fedora Core 1/x86
- Fedora Core 2/x86
- SLES 8/x86
- SUSE 9/x86
- SUSE 9.1/x86
- Windows (2000 and above)
- Mac OS X

For this article we are going to download the Red Hat 9.0/x86 version.There are two different ways you can download the beta-either one large zip file with everything in it or as individual RPM files.

The easiest way to install the Mono beta is to download the zip file that contains all of the RPM files that you need. Once you have unzipped the RPM files, you may try to install each of the individual packages by double-clicking each package to launch the installer, but there may be dependencies or other applications that need to be installed first. The easiest way to install all of the RPM files at once is from the command line.

Make sure you are logged in as the root user and then use the command line below to install all of the packages at once:

```
rpm -Uvh --nodeps *.rpm
```

If you already jumped ahead and installed a couple of the packages or if you already had some of these packages installed, you can force the installation of all of the Mono packages using:

```
rpm -Uvh --nodeps --force
*.rpm
```

Once you have installed all of the RPM packages, we need to go do some testing to make sure that all of the Mono components are running. Mono has its own lightweight Web server called XSP which was written using C# and ships with a number of test pages you can use to determine if the Mono beta is installed and working.

To start the XSP Web server, open a terminal and run:

```
mono /usr/bin/xsp.exe --port
80
```

With the Web server up and running, you can now view the test .ASPX pages that ship with the Mono beta by navigating to http://127.0.0.1/test/ where you should see the default Web page shown in Figure 1 (see next page).

Novell ships version 1.0 of Mono

Novell in early July released open-source software for programmers who want to use Microsoft's development tools to write applications on non-Windows operating systems, including Linux.

The Mono Project is intended to bring the ease of use that marks Microsoft's .NET tools to developers creating software for non-Microsoft operating systems, said Mono founder Miguel de Icaza, now vice president of development at Novell. Novell gained stewardship of Mono when it acquired open-source software company Ximian last year.

"Mono helps developers focus on what they are doing rather than the nitty-gritty details of the platform they are working with," de Icaza said.

As part of the release of Mono 1.0, Novell has started a Web site with resources and information for Mono developers.

The Mono 1.0 software includes a compiler for the Microsoft-created C# programming language and other .NET-based tools for speeding application development, including ASP.NET for building Web applications and ADO.NET for accessing databases. Mono also includes versions of the .NET "runtimes," software needed to run .NET applications, for Linux, Unix and Apple Computer's OS X.

De Icaza started the project three years ago, taking advantage of the publication of the C# language and other core pieces of its Visual Studio .NET development tools, including common-language runtime. Microsoft had submitted C# and related development technology to Ecma International for standardisation.

Like Microsoft's Visual Studio .NET development tool, Mono enables programmers to write software code with different languages, including C#, Java, Python, Visual Basic, and Jscript.

Microsoft originally released its first version of Visual Studio in 2002, well ahead of the release of Mono, which had delays in its delivery schedule before being released in a test version earlier this year. With the next version of Mono, de Icaza said, he thinks that the Mono Project will be able to more closely track updates to the .NET software, he said.

"We're already working on .NET 2.0 features," de Icaza said. Microsoft expects to deliver .NET 2.0 in the first half of next year with Visual Studio 2005. By contrast, many aspects of Longhorn, the next full upgrade of Microsoft Windows, are still unsettled, he said. "For Longhorn, we're not touching it yet."

Novell is so committed to Mono that it has stopped using the C++ language to develop two of its products--iFolder and ZenWorks--in favor of Mono.

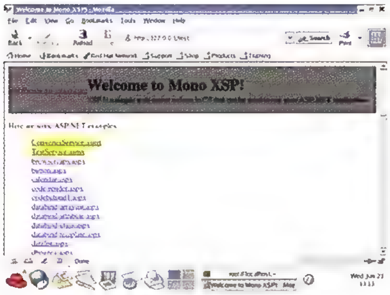
Although the Mono Project promises to simplify the process of building cross-platform applications, Microsoft does not view Mono as a competitive threat to its own development tools business, said John Montgomery, director of marketing for Microsoft's product division. Microsoft's .NET Framework, the runtime software needed to run and build .NET applications for Windows, is very thorough and can address many different application development needs, he said.

"Mono has taken a small subset of .NET and cloned--and it's unclear how much they've cloned and how good it is," Montgomery said.

—*Martin LaMonica*

There are a number of sample .ASPX pages that you can use see how Mono handles different ASP.NET features, including user interface controls, code-behind and different techniques for data binding and some sample Web services and documentation.

Now that you have confirmed that Mono is installed and that the Web server is up and running, we can start testing out your own .NET applications. One of the RPM files in the zip file provided with Mono Beta 3 contains a copy of MonoDevelop, an open source IDE that you can use to develop applications based on the Mono framework. You can find the Mono Develop icon in the program menu under Programming > More Programming Tools > MonoDevelop.



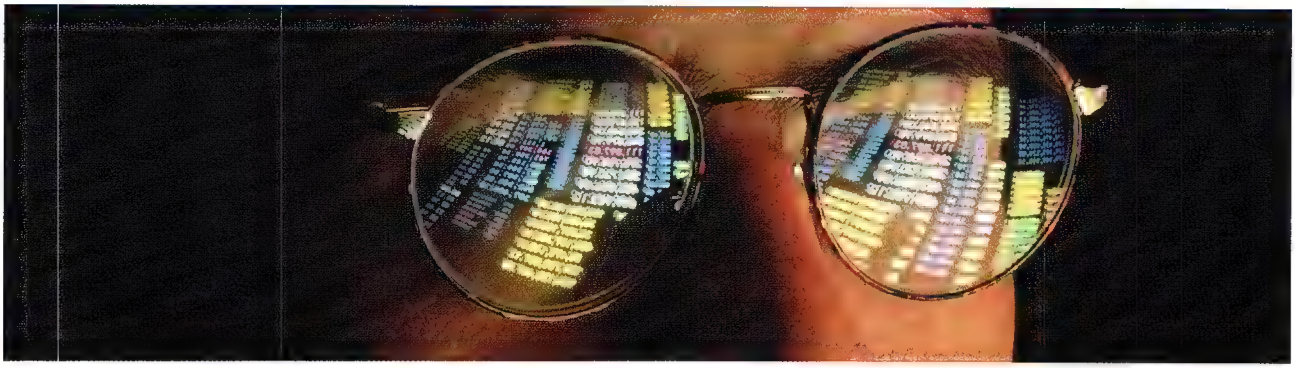
Welcome to Mono XSP!

🟩 Figure 1: Sample .ASPX pages.

Once you are in the MonoDevelop IDE, you can use the import option under the File menu to import a Visual Studio project and then compile and run it from within the MonoDevelop IDE. And don't forget, Mono supports C# but not VB.NET, so make sure any apps you want to test are written in C# before attempting to port them over.

Whether you are a Linux developer who is looking for a new toolset or a .NET developer looking for a new platform, Mono has a little bit of something for everyone.

ROBUST OBJECTS <u>AND</u> ROBUST SQL



A RARE SIGHT IN DBMS PRODUCTS

For your next generation of applications, move to the next generation of database technology.

Caché is the *post-relational* database that combines high-performance SQL for faster queries <u>and</u> an advanced object database for rapidly storing andaccessing objects. With Caché, no mapping is required between object and relational views of data. That means huge savings in both development and processing time.

Applications built on Caché are massively scalable and lightning-fast. Plus, they require minimal or no database administration.

More than just a database system, Caché incorporates a powerful Web application development environment that dramatically reduces the time to build and modify applications.

The reliability of Caché is proven every day in "life-or-death" applications at thousands of the world's largest hospitals. Caché is so reliable, it's the leading database in healthcare – and it powers enterprise applications in financial services, government and many other sectors.

We are InterSystems, a specialist in data management technology for twenty-five years. We provide 24x7 support to four million users in 88 countries. Caché is available for Windows, OpenVMS, Linux and major UNIX platforms – and it is deployed on systems ranging from two to over 10,000 simultaneous users.

InterSystems
CACHÉ™
Make Applications Faster

InterSystems International Corporation, Level 27, 363 George Street, Sydney NSW 2000. Phone: (02) 8243 0300 Fax: (02) 8243 0301.

Try a better database. For free.

Download a free, fully-functional, non-expiring version of Caché or request it on CD at <u>www.InterSystems.com/robust3</u>

ISC019

Migrating legacy applications

If your business has ancient applications chugging away on Windows NT or even a mainframe, here are our tips for smoothing the transition to a new system.

By David McAmis

As the cost of maintaining custom legacy applications increases, there are now a number of low-cost application servers that can be used to migrate these applications to other platforms and lower the total cost of ownership (TCO). In this article, we will take a look at the steps required to successfully migrate a legacy application to a more cost-effective platform for both development and maintenance costs.

1 Documenting the existing system

If we were to look at the software development lifecycle, a key part of the process is actually gathering requirements from users on the functionality that they need in the application you will eventually build. In the case of migrating a legacy application, the process is slightly different, as you will need to perform analysis on the existing system to understand how it works, who uses it, what they use it for.

A good starting point for gathering this information is in the existing documentation for the system you want to migrate. This documentation could take the form of the original specifications for the application, as well as the systems design and documentation produced once the application was completed. Often this information will be missing or incomplete with legacy applications, as the application most likely was developed quite a long time ago.

You may also find crucial information in other forms of documentation, including guides, manuals, tutorials, and training materials that end-users may have used. Most often this type of material will provide background information on the functionality exposed to end-users but may not provide details of how the underlying processes work.

For this part of the analysis, you will actually need to review and document the existing code. This may be difficult, depending on the legacy platform you are working with. For example, if you are migrating an AS/400 application written in RPG, you will definitely require some assistance from an experienced RPG programmer if you do not have those skills internally.

This can be an expensive part of the analysis process, as you may need to hire a resource to perform this analysis for you, but it is a vital part of the process, especially if the application has been running and maintained over a long period of time. There may be undocumented code or fixes that are critical to the application and that have not documented anywhere else.

Another key area to examine is how the data in the legacy system is stored. Is it stored in flat files, files, or tables? What fields are included in those files/tables and what indexes are in use? Also, perform a detailed analysis of any server processes that are running that may be related to the data—for example, if there is a nightly process that runs across a file and updates it from other systems

2 Requirements analysis

The next step in the migration process is to perform a detailed requirements analysis to determine what features are required in the new application. At a very basic level, the features and functionality that are in the legacy system should be included and you should perform interviews with the users and system administrators to see what features are required.

This area of the migration is frequently where the cost and scope of the project can increase if you do not manage this process. Often, the move from a legacy system to another platform is driven by the request for new features that a legacy application can't provide or where the cost of providing these features would be prohibitive.

In this situation, users will often use this exercise to include features that they may have wanted in the past that have been too costly or cumbersome to implement and may blow the scope of the migration project. A key point to emphasise to users is that the first step is a migration of existing features and functionality to a new platform.

The most effective way of cutting over to a new platform is to ensure that the application supports the existing features and functionality first before adding anything new. It is tempting to add features with the mentality of "while we are at it" but this can be a point of failure if those new features slow or halt development of core functionality that the business requires.

Seven deadly sins of integration

Today's mantra is that companies need to integrate different parts of their business in real-time, or else important information such as pricing, inventory levels, or financial data will remain stuck in a quagmire of incompatible systems.

This all makes sense, but there are many complex issues to consider before moving to this next phase of integration: business value, cost, leveraging existing investments, time to market, and total cost of ownership, to name a few.

As organisations adopt this new business model, they can make the transition more smoothly by avoiding the seven deadly sins of integration:

1 If it ain't broke . . .

Some companies are scared off by the cost of integration and perceive the current industry push as hype. Yet keeping silo architectures alive—where disparate systems and applications are disconnected and repeated across multiple business units—drives up costs due to inefficiencies. We recently worked with a company with multiple CRM, ERP, and other systems amassed through acquisitions. We helped them establish a shared service centre to blend IT operations that increased customer satisfaction, presented a unified company brand, and boosted company-wide purchasing power.

2 Let's rewrite from scratch

It's appealing to rip out what you have and write new applications, perfectly designed for your organisation with flashy, modern interfaces. But it's not always practical when your company is faced with critical, aggressive time-to-market demands. What is new today is obsolete tomorrow, so you get trapped in a cycle of playing catch-up—without ever realising the value of the newly rebuilt systems.

3 I already have an EAI solution

While traditional EAI with its hub-and-spoke architecture works for integrating multiple applications, it is likely to be proprietary. EAI combined with a standards-based architecture such as Web services creates a broader service-oriented architecture that avoids vendor lock-in and its associated costs.

4 There's no way out

If you're running a good, solid legacy system that is too expensive to maintain, it may feel like there is no way out. But with the appropriate tools and technology, it is possible to migrate into the world of open standards. Some platform vendors offer solutions that reuse an application's business logic, so replacement can be the right choice in many cases.

5 I don't need to outsource services

If you have IT expertise within your company, it may make sense to handle the integration yourself. But consider consulting an expert to create an integration backbone so your internal IT team can focus on building business applications. The expert can also mentor your staff so they can maintain the technology foundation themselves.

6 Give me one of everything

The range of software choices today can be overwhelming. Go for best-of-breed products and select the software that works best for you. Some vendors such as BEA bundle a suite of software on top of their flagship technology and push it as a single package. You are better served by an adaptable integration foundation based on industry standards, and then select the other parts of your integration solution.

7 Integration is not for me

Integration used to be for the big guys, but that's no longer the case. Today, small and medium businesses deal with a plethora of disparate systems, inefficient operational processes, and the need to open applications up to partners and customers. Solutions are rapidly becoming affordable and packaged specifically for small and medium businesses, so SMEs can benefit from integration as well.

Even for companies that can successfully sidestep these seven common sins of integration, it's important to choose the right integration foundation. True business integration involves modelling, monitoring, and managing business processes as well as integrating people, processes, systems and data with customers and partners.

That means integrating across heterogeneous environments, which requires an open, any-to-any environment. Companies should consider selecting an integration platform from a vendor with a multi-platform strategy—such as IBM—rather than a mono-platform solution, such as Microsoft.

Software that is open, approachable and agnostic enables customers to cut costs and increase revenue and business opportunity. That winning combination advances integration and, ultimately, business on demand. And that's what separates a faddish marketecture from a powerful vision.

Nicolas Jabbour is CTO and VP of Business Development at Prolifics.

A good way to limit the scope of the migration project is through the development of "use cases" that detail how the legacy system is currently being used. You can then use these use cases to design your new application and ensure that the exact functionality offered in the legacy system is included in your application.

3 Platform selection

With an understanding of how the application is being used and the code behind it, as well as the user requirements, you can then select a platform for the migrated application. With legacy applications, this decision was most likely made on a consolidated platform where the application logic and data both resided on the same platform. With modern applications, the application logic and data are usually separated into two distinct components, so you will need to decide on both an application and database platform.

The major decision to be made on the application platform is whether the migrated application will be a "fat" client application, installed locally on each user's computer or whether it will be a Web-based application that the user can run from their browser. While current trends lean toward replacing desktop applications with Web-based versions, there may be some functionality that can't be easily replicated in a Web application.

For example you could have a mainframe application that performs complex statistical analysis as you change program parameters and then returns a result set to the user, with calculations running 4-5 minutes each time. In a Web environment, this may not be possible as the Web browser may time out before the result set could be returned. So in this case you have two choices—you can either provide this functionality using another method that suits Web applications (eg, posting the parameters to a form and then have the user visit another page to retrieve the results) or consider deploying a fat client application, which would allow you to have similar functionality.

In addition to the fat vs thin debate, there is also a development language to be selected. You may currently have a corporate standard development environment that you can use or you could use this as an opportunity to establish a new project. When considering a programming platform, make sure that you include the cost of servers in your evaluation. Will you need to buy additional servers or can the application be hosted on current hardware? Will the application scale and meet your needs in the future or will you need to add additional servers to support users? All of these issues should influence your decision.

From the data side, you will need to select a database platform that can support the application you plan to create, as well as hold any existing data you want to migrate over from the legacy system. The database platform should lend itself to the application platform you have selected—for example, if you are planning on creating a .NET Web application, there are definitely advantages to using a database that has .NET integration, which include SQL Server, Oracle, DB2, and others.

You should also look the amount of behind-the-scenes processing that was contained within the legacy system and make sure that your selected database platform can support the required load.

4 Design, development and deployment

Finally, with the system requirements ironed out and the application and database platform you can start the detailed design and development phase of the migration, where you actually create the detailed design document for the system you are migrating to and start development.

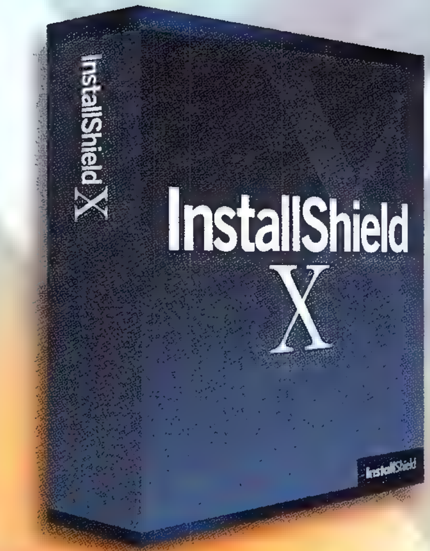
Another key to a successful migration is ensuring that you do a thorough detailed design that encompasses the current functionality in the legacy system and user requirements.

Users should be able to take the screen designs from your detailed designs and walk through the tasks they would normally perform in the legacy system you are replacing. When looking at the detailed design, do a check by walking through the use cases you originally created from the legacy app and ensure that there the new application supports all relevant cases.

Also during the development phase keep the development team focused on the detailed design you have created and perform frequent test migrations of data to ensure that the new application is ready to go live when you are ready to cutover. Before the final cutover point you should run both systems in parallel to ensure that there are no critical gaps in support for the business. It may be tempting to go with a straight cut-over from the old to new system, but there is a risk that the new system may not provide the functionality required or may have critical errors that prevent users from using it.

It is best to run the systems in parallel through a normal processing cycle—for example, if you are replacing an order entry system that performs month-end processing, make sure that you run both systems through the entire process. Once you are satisfied that the new application is fully functional, you should leave the legacy application and platform up and running for the next few months. And before you finally pull the plug, make sure that you have a complete backup of all of the relevant data and applications and that you have a way to access this information when required.

So with a bit of careful planning you can migrate legacy applications to a platform with a lower cost of ownership, both from the support and development perspective. ⬟


InstallShield X
InstallShield
X
InstallShield

Protocol independence in a service-oriented architecture

Hard-wiring protocols into your system is a recipe for disaster if things change. Here's how to build protocol-independent systems instead.

By Jeff Hanson

Because of the way most distributed systems encapsulate the protocol details of inter-process communication, it can be difficult to update your system when protocols change. You should build protocol independence into every aspect of your system.

Designing an effective distributed software system becomes especially tricky when considering the fact that services and processes might need to interact with each other across multiple protocols.

Most distributed systems encapsulate the protocol details of interprocess communication within the processes or services that use each given protocol. Either that or the details are embodied within some manager class charged with handling communications for a given protocol. Both mechanisms raise the need to redesign, reprogram, recompile, and test whenever a new protocol is needed.

Depending upon the amount of coupling between the protocol-dependent code and the services themselves, there is usually a fair amount of interdependencies created, leading to potentially difficult couplings to deal with when the underlying communication infrastructure changes.

Designing for protocol independence

A service-oriented architecture (SOA) depends on a minimum amount of interdependencies between services. This loose-coupling must propagate all the way to the protocol layer of the architecture. Thus communication infrastructure used within an SOA should be designed to be independent of the underlying protocol layer.

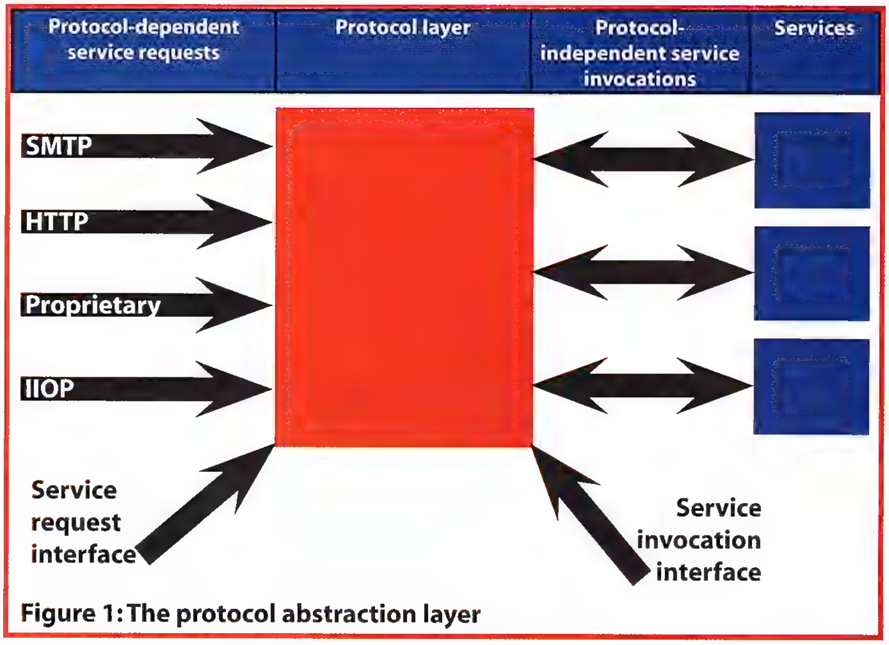

Protocol-dependent service requests
Protocol layer
Protocol-independent service invocations
Services
SMTP
HTTP
Proprietary
IIOP
Service request interface
Service invocation interface

Figure 1: The protocol abstraction layer

Code reusability becomes one of the most important benefits derived from a well-designed protocol-independent communication framework, since the code that uses the framework will not be tied to the protocols used. Transparent replacement of the underlying protocol is also a very important benefit derived from a protocol-independent design. This facilitates deployment of a new protocol without the typical redeployment headaches such as server restarts and frustrated clients.

Patterns for protocol independence

- **Business delegate.** Hides underlying implementation details of a business service, such as lookup and access details. For example, a Java Bean wrapper that aggregates multiple objects and presents an easily understood business interface.
- **Remote proxy or surrogate.** Acts as a stand-in for objects which exist in a different tier. For example, RMI's stub class design.
- **Adaptor or wrapper.** An adaptor provides transparent access to disparate objects or services by converting requests and possibly responses from one interaction interface to another. An example is Java's Portable Object Adapter.
- **Broker.** Decouples business tier objects from objects and services in other tiers, for instance CORBA's ORB environment.
- **Factory.** Instantiates objects at run-

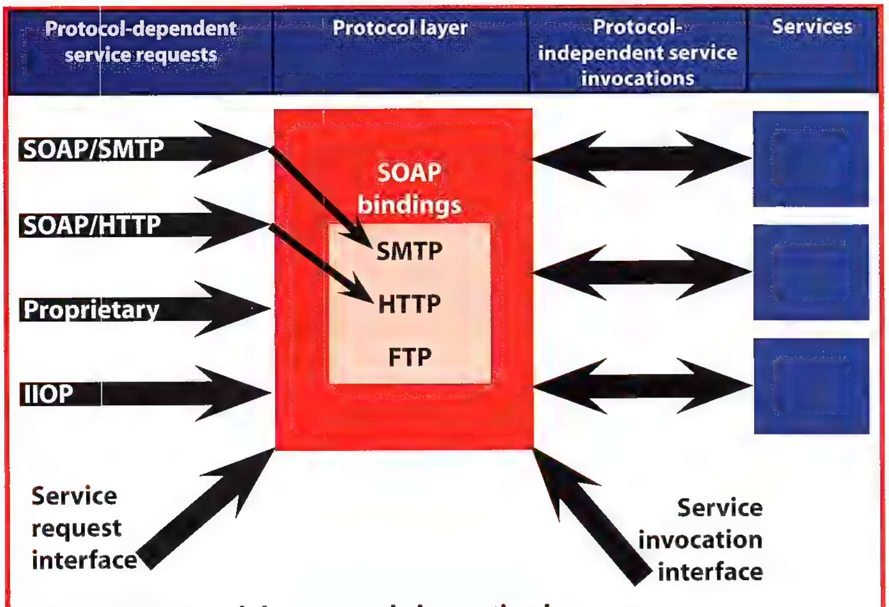

Protocol-dependent service requests
Protocol layer
Protocol-independent service invocations
Services
SOAP/SMTP
SOAP/HTTP
Proprietary
IIOP
SOAP bindings
SMTP
HTTP
FTP
Service request interface
Service invocation interface

Figure 2: SOAP and the protocol abstraction layer

time based on dynamic configurations. Factories are often instantiated objects designed around the Strategy pattern or Bridge pattern. An example is RMI's custom socket factory environment.

Protocol abstraction layer

One of the most important pieces of technology in a protocol-independent framework is a protocol abstraction layer. Among other benefits, a protocol abstraction layer allows access to services from multiple, simultaneous protocols, hides protocol details from the service developer, and facilitates transparent protocol replacement. Figure 1 illustrates this concept.

Protocol independence with SOAP

The Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol for exchanging information in a decentralised, distributed environment. SOAP supports Remote Procedure Call (RPC) style of information exchange and message-oriented style of information exchange. SOAP is designed around a number of loose-couplings including protocol independence, language independence, platform independence, and operating system independence.

The SOAP specification defines the structure of an XML document that is to be used to exchange data between two entities over a variety of other protocols.

The SOAP specification, version 1.2, defines a binding framework explaining the responsibilities for transporting a SOAP message from one node to another. This makes SOAP processors protocol-agnostic and protocol-independent. Thus, SOAP version 1.2 messages can be transported across HTTP, SMTP, or any other protocol for which a binding conforms to the binding framework.

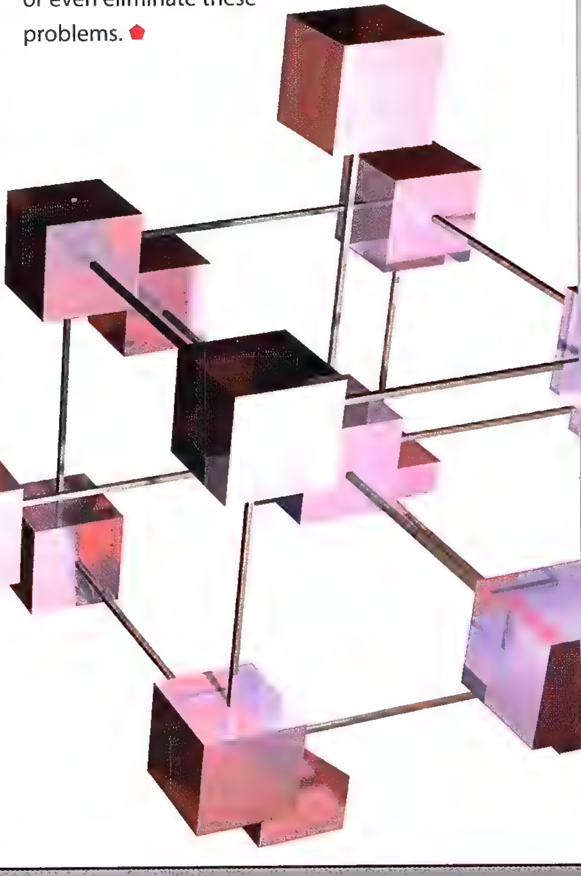
The SOAP binding framework:

- Declares the features provided by a binding.
- Describes how the services of the underlying protocol are used to transmit SOAP message infosets.
- Describes how the services of the underlying protocol are used to honour the contract formed by the features supported by that binding.
- Describes the handling of all potential failures that can be anticipated within the binding.
- Defines the requirements for building a conformant implementation of the binding being specified.

The SOAP binding framework can be easily integrated with existing protocol abstraction layers by providing bindings for protocols that are already handled in the legacy code. For example, let's look at Figure 2, which shows the protocol abstraction layer integrated with the SOAP binding framework.

As shown in the diagram, the service-request interface and the service invocation interface do not need to change. This allows for easy migration without runtime disturbances or interfacing code changes.

Designing an effective distributed software system can become complex when services and processes need to interact with each other across multiple protocols. The introduction of each new protocol also typically introduces the need to redesign, reprogram, recompile, and retest the system. However, a service-oriented architecture built around a properly designed protocol abstraction layer using standard industry patterns can effectively reduce or even eliminate these problems.



Designing for distant shores part 1: technical

Australia's limited population of IT users means that most developers with commercial ambitions want to sell their products overseas, but what are the design and marketing challenges involved in doing so? In the first of a twoËpart series, Angus Kidman examines the technical challenges of building applications for the international market.

Selling to the international market is the obvious next step for software developers who have come up with a great product or technology and seen it achieve some success in Australia. Unfortunately, getting your brilliant piece of work onto the world stage requires more than just technical nous and the ability to get your Web site listed on Google.

Broadly speaking, the challenges which face developers seeking to expand overseas fall into two categories: technical and commercial. Most developers are aware that selling outside their home country can be difficult. "One of the bigger issues for small companies is that it seems very daunting to go international," notes Shane Williamson, partner manager for the development program at mobile network supplier 3 and a veteran observer of Australian developers.

While the commercial problems are often well recognised (and will be examined in detail in part two of this article), developers aren't always as cognisant of the fact that international success begins at the design and coding stage. "I've seen a lot of attempts fail because they don't bother to do their research and understand the reasons to take the software to other markets," says Williamson.

"If you've got a product you're going to take to market, you need to think of who the end user is," agrees Chris Eldrige, director of software engineering solutions at Dialect Solutions, a Brisbane-based company whose payment systems have been sold internationally. "Software engineers who are caught up in the technology can easily forget that."

"Developers don't always do a good job of explaining how a product is relevant to business," adds Graham Jones, head of Integrated Research. While IR's Prognosis system management programs were developed in Australia, the company always intended to sell them overseas, and spent more than two years on development and research before launching their first version.

"At the technical level, there were a lot of early decisions to make about strategy," says Jones, whose advice to budding international entrepreneurs is succinct: "Don't bite off more than you can chew."

Differentiation isn't always a matter of purely technical skill. Gerard Caré, managing director for G+D Computing, whose Strand7 engineering product derives 50 percent of its sales offshore, first spotted an obvious gap in the market 15 years ago: "The bigger products were more difficult to use." Technical competence was an essential feature for an engineering package, but ease of use and consistency made it more appealing to users long-term.

Speaking in tongues

Perhaps the most visible challenge in attacking other countries is the language barrier: how easy will it be to develop a version of your product in French or Mandarin? "The biggest challenges historically have been the production of software that could provide double-byte capabilities for languages such as Chinese, Thai, and Japanese," says David Steny, business development manager for Tokuii, which develops relationship management software.

While this can seem tricky at first, the rise of Unicode as a basic standard has made matters rather simpler. "Make sure you support Unicode character sets from day one," advises Phil Morle, chief technology officer for Sharman Networks, whose Kazaa P2P tool has certainly succeeded in garnering international attention, albeit not always in a complimentary way. "Doing this later is complex and not doing it at all means you are limiting the regions you can localise your product to."

"If I was doing it all from scratch, I'd do everything Unicode, but that wasn't an option at the time," notes G+D's Caré. However, he also points out that not all overseas buyers are concerned about translation.

"In the engineering world, it's a highly educated target market. In most Western countries, language hasn't been an issue for us," he says. Even in China, many university students prefer software to be in English.

Jack Andrys, CEO of WebSpy, a listed company whose Internet monitoring software is sold across Europe, suggests that pursuing multilingual software is only worthwhile once you have a very large user base. "English has continued to be more and more acceptable," he says.

"The chances are you are never going to get the budget allocation to translate and localise your application into Estonian or Zulu, but the aggregation of all those smaller markets is a pretty big market," adds Sharman's Morle. "If you have done the job right in separating the locale-specific elements of your application, it should be easy enough to create tools to let users do their own localising and contribute their efforts."

Remember that translation requires more than just word matching.

"Localisation is much more than just getting a translation service to give you copy in the target language," says Todd Martin, executive producer for Amnesia Group, which develops multimedia marketing content across Asia.

"The language used must fit the purpose and sound like it comes from a knowledgeable local." Martin notes this is less of an issue for highly technical software: "The difficulties of marketing and language are probably not as critical for software developers as for designers. Software users are focusing on whether the product suits their needs rather than the nuances of communication to differing demographics."

If you do decide to develop multiple language versions, then you'll need to put in some extra effort in even if—as seems likely—you rely on third parties for the actual translations.

"Don't assume anything," says Morle. "Don't expect a word in one language to be the same size, or go in the same direction in other languages. Don't expect a comma in a string of numbers to mean the same thing someplace else."

The culture of abstraction

Outside of language, the next most obvious challenge for many international products is date formats: Australia and most European countries now use DDMMYY, but the US stubbornly persists with MMDDYY. While these differences are handled fairly well in most modern operating systems, be wary of making assumptions. For instance, the US military uses "international" dates in the YYMMDD format.

This underlines the important point that software often requires subtle process changes. "A key component is how that software interacts with other systems—the business logic often needs to be changed," says Richard Dowling, Rational technical manager for IBM ANZ. "You don't want two different applications. You want to use the same components."

"The expectations of scale will be different in larger markets," says Doug Bertinshaw, chief operating officer for Integrated Research. A product that works well in a typical Australian business may struggle as user loads soar.

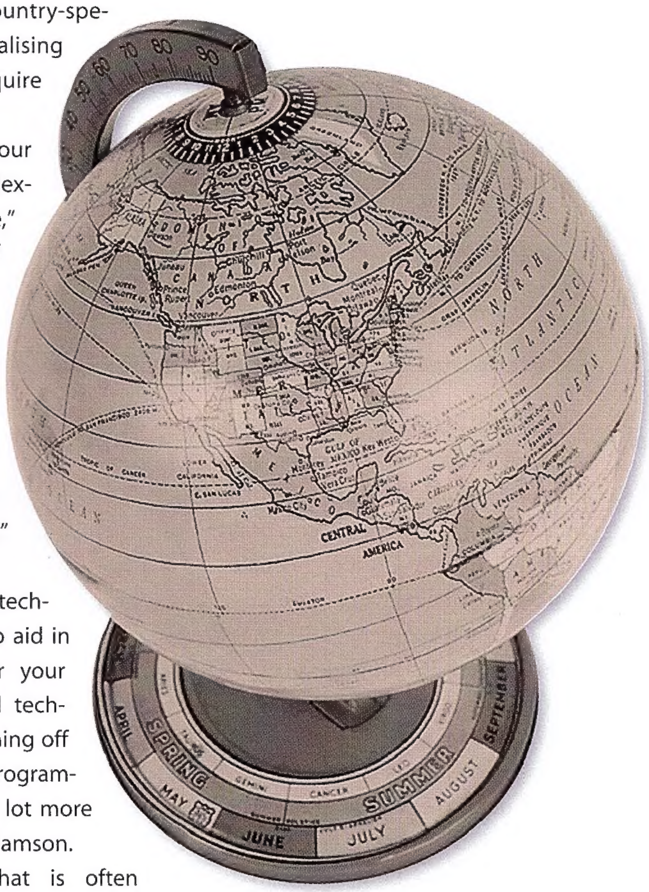
Generally speaking, the best way to face these challenges is to plan your software to enable flexible redeployment for different markets. "Make sure that you only need to do things once. This is the number one rule," says Morle. "Keep application logic away from country-specific resources so that localising your product does not require re-writing the application."

"You've got to build your product so that's it's very flexible in terms of architecture," says Grant Straker, chief technical officer of Straker Interactive, whose content management system has been sold through New Zealand, Australia, and Europe. "It's extremely difficult to retrofit a system to fit a new region or market," concurs Rational's Dowling.

Working with established technology frameworks can also aid in this process. "If you tailor your product on a standardised technology, and don't go branching off into some weird abstract programming language, you'll get a lot more global traction," says 3's Williamson.

One important area that is often overlooked is building extensive troubleshooting features into your architecture—

> *"Make sure that you only need to do things once. This is the number one rule."*



How to evaluate an offer of a job abroad

Overseas assignments can be a lot of things: exciting, intimidating, fulfilling, educational, career-enhancing, career-limiting. As with any decision, the first step in deciding whether to accept such a position is an investigation of the pros and cons. You should evaluate an overseas assignment the same way you would any local promotion, transfer, or new job.

These are just a few questions that you'll need to answer:

- Will you learn new marketable skills?
- Will you be recognised for your accomplishments?
- Will you earn raises and promotions faster?

Once you've answered these questions, review the issues particular to overseas assignments:

Speak to the people in your organisation who have completed an overseas assignment, if any—has their career advanced; did their personal life suffer? Find out what they took into consideration in making their decision to go abroad.

Take a hard look at what the overseas role would do for you professionally. Find out if it's a position that will be available at home when you return, and what the next step would be on your return.

Ask about your say on the **duration of the job**—what if you need to cut it short due to a family issue or because you realise you can't stand the new role? On the other hand, what if you find that you love the assignment and you don't want it to end? An IT executive I've known for years is finishing his second overseas assignment. In both cases, he extended the positions beyond the initial term. In both cases, he would have extended them again if he could have.

Get all the facts on the **compensation plan**—is it an "expat" or "local" approach? I recommend that you be sure you'll be sent as an expat, and that you negotiate in advance that you'll continue on the expat pay plan. What you want to avoid is the pressure to be paid in the local currency at the local prevailing wage. When you first head overseas, your company sends you because they don't have that talent available locally. Over time, you'll likely be grooming resident IT workers to take over. Here's the rub: halfway through your overseas assignment, the locals will be half-trained, so you will have lost half your value. This is when your company will try to convert you to the local pay plan. You'll be in no position to negotiate pay plans at that point so you need to get that commitment up front and in writing.

Spend lots of time examining **family issues**; this is often the key element in whether an abroad experience is successful. The younger your kids are, the easier it usually is. Do you want your children educated in an international or local school, and can you afford the expense associated with your preference? What will this do to your partner's career?

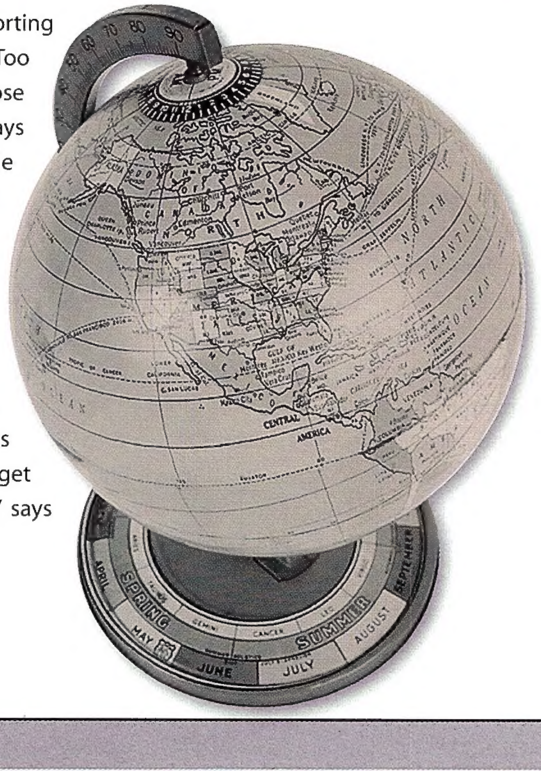
Is there a **guarantee in taking the offer**—does any promotion or transfer come with a guaranteed fallback scenario should the position not work out? Typically the answer will be no, because no one gets a guarantee in any domestic position. But it's still worth trying to negotiate if the new role is tied to a new business endeavour, for example.

—Peter Woolford

something that can make supporting offshore users much easier. "Too often, we don't think about those things until after the event," says Dialect Solutions' Eldridge. "People might be ditching your product and you won't know."

"You must build a high-level of supportability in your product," agrees IR's Jones.

A related issue is the need to have a good base of beta testers, and ideally to have them across the globe. "You are only going to get good results if you have testers," says Williamson.



Dealing with all these issues can take considerable time and resources—and bear in mind that you haven't even started trying to sell the application yet. However, careful planning can eliminate much of the stress. As Dowling points out: "If it's a well-developed system and you understand what region you are targeting the application for, it won't be a major exercise."

In the next issue of Builder Magazine, *we'll discuss how to pick your overseas target markets, plan a sales strategy, and ensure that you're not overwhelmed by establishment or support costs.*

BUILDER • CONFERENCE

5 - 7 October, Star City, Sydney

Register now and save up to $300!*

About the Conference

Join your peers in the Builder AU community as they meet face-to-face at the inaugural Builder Conference, the first platform-neutral event of its kind in Australia. Produced by developers, for developers, the Builder conference is your opportunity to share ideas and hear about the latest trends and issues in application development, free and open source software, .NET, Java, SOA, Web services, mobile devices and more. Register now at www.builderconference.com.au

Featured Speakers include:

Richard Stallman

Richard Stallman is the founder of the Free Software Movement and lead developer of the GNU operating system. Stallman or RMS as he is also known as wrote the first extensible Emacs text editor at MIT in 1975 is the also the principal author of the GNU Compiler Collection, GNU symbolic debugger (gdb) and GNU Emacs.

Ken Arnold

Ken Arnold, a legend in the software development industry, is one of the inventors of Jini, the designer of JavaSpaces and contributor to the design of CORBA, the curses library package and co-author of rogue.

Allan Paull

Professor Allan Paull is Program Manager for the HyShot Project at The University of Queensland. Allan has achieved significant contributions toward supersonic combustion ramjet (scramjet) research and development in Australia over the past ten years.

For more speakers and a complete agenda refer to builderconference.com.au

Platinum Partner

msdn
Microsoft® Developer Network

Gold Partners

ca
Computer Associates®

bea

Intel® developer services

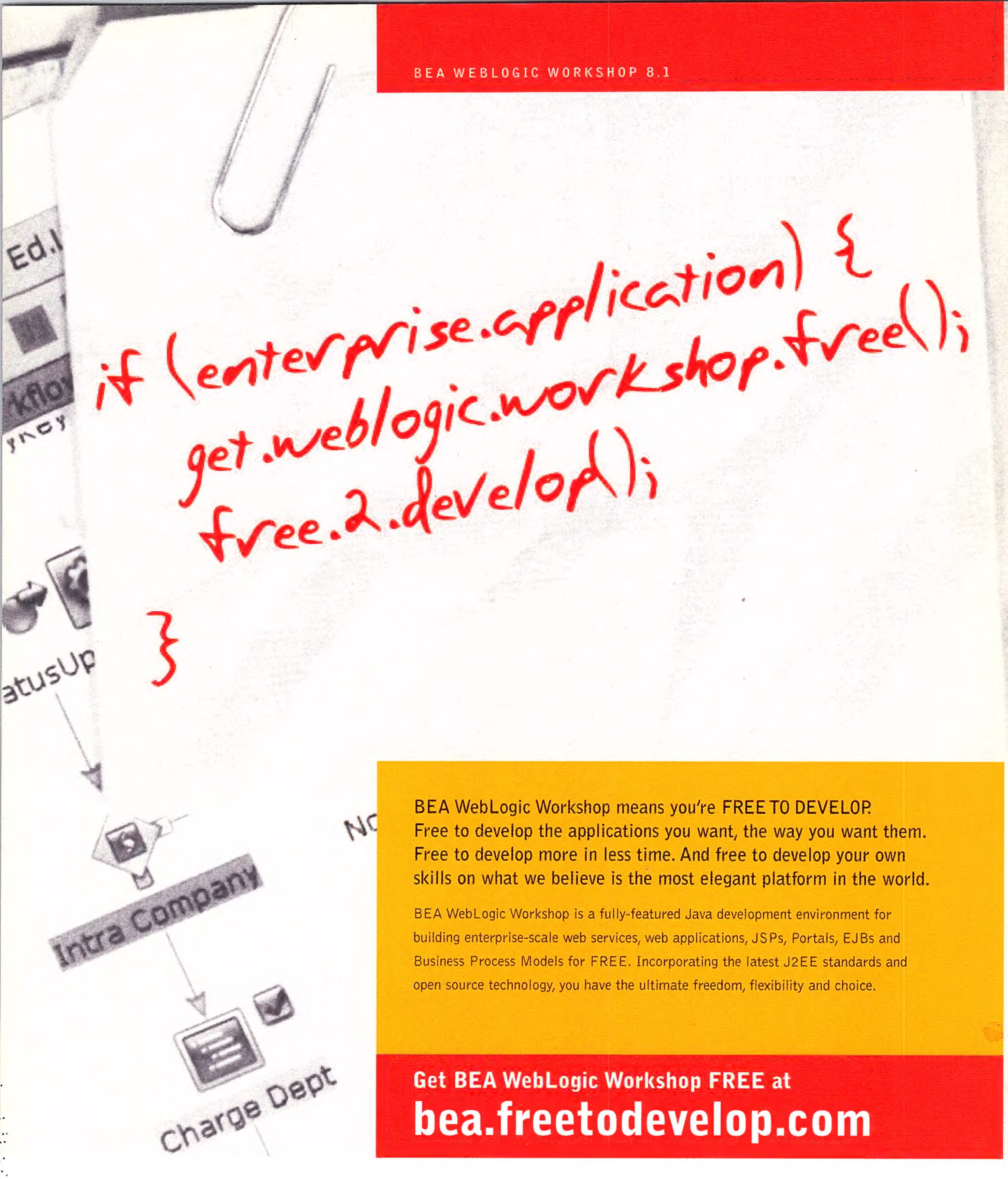
Brought to you by

BUILDER • AU
builderau.com.au

**Early bird discount for Builder AU community members is $995 for two-day conference. Optional 3rd day workshop also available. Early bird valid only until August 14, 2004. Register now!*

For more information or to register visit www.builderconference.com.au

BEA WEBLOGIC WORKSHOP 8.1



BEA WebLogic Workshop means you're FREE TO DEVELOP. Free to develop the applications you want, the way you want them. Free to develop more in less time. And free to develop your own skills on what we believe is the most elegant platform in the world.

BEA WebLogic Workshop is a fully-featured Java development environment for building enterprise-scale web services, web applications, JSPs, Portals, EJBs and Business Process Models for FREE. Incorporating the latest J2EE standards and open source technology, you have the ultimate freedom, flexibility and choice.

Get BEA WebLogic Workshop FREE at

bea.freetodevelop.com

bea®